Curriculum

Department of
Computer Science and Engineering

<u>Undergraduate</u>
Session: 2024-2025

<u>Graduate</u>
Session: 2024-2025

Shahjalal University of Science and Technology
Sylhet, Bangladesh

Published by:
Office of the Registrar
Shahjalal University of Science and Technology (SUST)
Sylhet 3114, Bangladesh

Syllabus Publication Committee:

Chairman:

Members:

Secretary:

Cover Design:

Shahjalal University of Science and Technology
Sylhet-3114, Bangladesh
PABX: +88-0821-713491, 714479, 713850, 716123, 715393
Fax: +88-0821-715257, 725050
Website: www.sust.edu
E-mail: registrar@sust.edu

Contents

FACULTY LIST (Current)

Sl. No.	Full Name	Room No.	Cell Phone
Professors			
1.	Dr. M. Shahidur Rahman	333	01914930807
2.	Dr. Md. Reza Selim	604	01972357830
3.	Dr. M. Jahirul Islam	603	01770348185
4.	Dr. Mohammad Abdullah Al Mumin	604C	01711445110
5.	Mr. Md. Masum	604A	01919736248
6.	Dr. Md. Forhad Rabbi	604B	01844175805
Associate Professor			
7.	Dr. Husne Ara Chowdhury	603A	01712834619
8.	Dr. Sadia Sultana	603B	01911089612
Assistant Professors			
9.	Mrs. Mahruba Sharmin Chowdhury	720	01917566699
10.	Ms. Ayesha Tasnim		
11.	Mr. Md. Eamin Rahman	710	01677014633
12.	Mr. Md. Saiful Islam*		
13.	Mrs. Marium-E- Jannat*		
14.	Mr. Md Mahfuzur Rahman*		
15.	Mr. Md Mahadi Hasan Nahid*		
16.	Mr. Enamul Hassan*		
17.	Mr. Moqsadur Rahman*		
18.	Mr. Summit Haque*		
19.	Arnab Sen Sharma*		
20.	Maruf Ahmed Mridul*		

Sl. No.	Full Name	Room No.	Cell Phone
Lecturer			
21.	Mr. A. K. M. Fakhrul Hossain	709	01309354539
22.	Mr. Shadmim Hasan Sifat	711	01571291835
23.	Mr. Md. Mehedi Hasan	721	01932981532
24.	Mr. Abdullah Al Noman	712	01601160217
25.	Mr. Ishtiaque Zahid	707	01715572978
26.	Mr. Abdullah Al Thaki	323	01516178219
27.	Md. Shymon Islam	324	01718550548
28.	Md Fahimul Islam Fahim	321	01893639181

*On Study Leave

Department of Computer Science and Engineering
Shahjalal University of Science & Technology
Sylhet-3114, Bangladesh
Curriculum for B. Sc. (Engg.) Program
Session: 2024-25

Vision Statement

The Department of Computer Science and Engineering, SUST intends to provide an excellent educational environment in order to develop professionals with strong technical and research backgrounds.

Mission

M1. To provide quality education in both theoretical and applied foundations of Computer Science and Engineering.
M2. To prepare highly skilled computer engineers, capable of doing research and also develop solutions for the betterment of the nation.
M3. To inculcate professional and ethical values among students.
M4. To support society by participating in and encouraging technology transfer.

Program Name: B.Sc. (Engg.) in Computer Science and Engineering

Program Educational Objectives (PEO)

The entity has set the following PEOs for the B.Sc. (Engg.) program in Computer Science and Engineering major:

PEO1. To provide students with a strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems and to prepare them for graduate studies, R&D, consultancy and higher learning.

PEO2. To develop an ability to analyze the requirements of the software, understand the technical specifications, design and provide novel engineering solutions and efficient product designs.

PEO3. To provide exposure to emerging cutting edge technologies, adequate training opportunities to work as teams on multidisciplinary projects with effective communication skills and leadership qualities.

PEO4. To prepare the students for a successful career and work with values and social concerns bridging the digital divide and meeting the requirements of local and multinational companies.

PEO5. To promote student awareness on life-long learning and to introduce them to professional ethics and codes of professional practice.

Mission Statement to PEO Mapping

Mission/PEO	PEO1	PEO2	PEO3	PEO4	PEO5
M1	X	X	X		
M2	X	X	X	X	
M3				X	X
M4			X		X

Program Outcomes (PO)

After graduation from this program in CSE, the graduates will be able to:

PO 01. Engineering Knowledge
Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO 02. Problem Analysis
Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 03. Design/Development of Solutions
Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety as well as cultural, societal and environmental concerns.

PO 04. Investigation
Conduct investigations of complex engineering problems, considering design of experiments, analysis and interpretation of data and synthesis of information to provide valid conclusions.

PO 05. Modern Tool Usage
Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 06. The Engineer and Society
Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 07. Environment and Sustainability
Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of need for sustainable development.

PO 08. Ethics
Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 09. Individual and Team Work
Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10. Communication
Communicate effectively on complex engineering activities with the engineering community and with society at large. Some of them are being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11. Project Management and Finance
Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12. Life Long Learning
Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Program Outcomes (PO) to Program Educational Objectives (PEO) Mapping

PO/PEO	PEO1	PEO2	PEO3	PEO4	PEO5
PO 01	X	X	X		
PO 02	X	X			
PO 03	X	X		X	
PO 04			X	X	X
PO 05	X	X			
PO 06			X	X	
PO 07				X	X
PO 08				X	
PO 09	X				X
PO 10			X		
PO 11			X	X	
PO 12					X

Graduate Profile:
Students graduating from the department of CSE, SUST should have gained the following attributes.

a. Intellectual skills in Science and Engineering
b. Practical and problem solving skills
c. Numeracy and analytical skills
d. Entrepreneurship and innovation skills
e. Communication skills
f. Interpersonal, teamwork and leadership skill
g. Self-management & personal development skills
h. Commitment to community, country and humanity

Semester wise Curriculum Breakdown:
One-semester credit hour represents one class hour or two laboratory hours per week. An academic semester represents 13 weeks of classes exclusive to final exams. Semester wise breakdown of the curriculum structure for 2022-23 session are shown below.

Department of Computer Science and Engineering
Undergraduate Program
Session 2024-2025

First Year: 1st Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0613 1133	Structured Programming Language	3	0	3.0	
CSE	0613 1134	Structured Programming Language Lab	0	6	3.0	
CSE	0541 1143	Discrete Mathematics	3	0	3.0	
ENG	0231 1101D	Effective Communication in English	2	0	2.0	
ENG	0231 1102D	English Language Lab 1	0	2	1.0	
MAT	0541 1101D	Coordinate Geometry and Linear Algebra	3	0	3.0	
EEE	0713 1109D	Electrical Circuits	3	0	3.0	
EEE	0713 1110D	Electrical Circuits Lab	0	3	1.5	
Total			**14**	**11**	**19.5**	

First Year: 2nd Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0610 1250	Capstone Project Work I	0	3	1.5	
CSE	0613 1237	Data Structure	3	0	3.0	0613 1133
CSE	0613 1238	Data Structure Lab	0	4	1.5	
PHY	0533 1203D	Mechanics, Wave, Heat and Thermodynamics	3	0	3.0	
MAT	0541 1203D	Calculus	3	0	3.0	
EEE	0714 1211D	Electronic Devices and Circuits	3	0	3.0	0713 1109D
EEE	0714 1212D	Electronic Devices and Circuits Lab	0	3	1.5	
IPE	0732 1206D	Engineering Graphics	0	3	1.5	
PHY	0533 1202D	Basic Physics Lab	0	3	1.5	
Total			**12**	**16**	**19.5**	

Second Year: 1st Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0613 2133	Object Oriented Programming Language	3	0	3.0	0613 1133
CSE	0613 2134	Object Oriented Programming Language Lab	0	3	1.5	
CSE	0613 2137	Algorithm Design and Analysis	3	0	3.0	0613 1237
CSE	0613 2138	Algorithm Design and Analysis Lab	0	3	1.5	
ECO	0311 2105D	Principles of Economics	3	0	3.0	
BUS	0411 2105D	Cost and Management Accounting	3	0	3.0	
STA	0542 2102D	Statistics and Probability	3	0	3.0	
IPE	0715 2108D	Computer Aided Drawing	0	3	1.5	
Total			**15**	**9**	**19.5**	

Second Year: 2nd Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0610 2250	Capstone Project Work II	0	6	3	
CSE	0612 2260	Web Technologies	0	3	1.5	
CSE	0613 2234	Introduction to Competitive Programming	0	3	1.5	
CSE	0613 2247	Theory of Computation	3	0	3.0	
CSE	0688 2291	Data Science	2	0	2.0	
CSE	0688 2292	Data Science Lab	0	3	1.5	
CSE	0541 2257	Numerical Analysis	2	0	2.0	
CSE	0541 2258	Numerical Analysis Lab	0	3	1.5	
MAT	0541 2204D	Complex Variables, Laplace Transform and Fourier Series	3	0	3.0	
EEE	0714 2201D	Digital Logic Design	3	0	3.0	0713 1109D, 0714 1211D
EEE	0714 2202D	Digital Logic Design Lab	0	3	1.5	
Total			**13**	**21**	**23.5**	

Third Year: 1st Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0612 3163	Database System	3	0	3.0	
CSE	0612 3164	Database System Lab	0	3	1.5	
CSE	0613 3135	Operating System and System Programming	3	0	3.0	
CSE	0613 3136	Operating System and System Programming Lab	0	3	1.5	
CSE	0613 3141	Software Engineering and Design Patterns	3	0	3.0	
CSE	0613 3142	Software Engineering and Design Patterns Lab	0	3	1.5	
CSE	0619 3193	Artificial Intelligence	3	0	3.0	
CSE	0619 3194	Artificial Intelligence Lab	0	3	1.5	
CSE	0714 3177	Microprocessor and Embedded System	3	0	3.0	0714 2201D
CSE	0714 3178	Microprocessor and Embedded System Lab	0	3	1.5	
Total			**15**	**15**	**22.5**	

Third Year: 2nd Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0610 3250	Capstone Project Work III	0	6	3.0	
CSE	0611 3230	Technical Writing and Presentation	0	3	1.5	
CSE	0619 3293	Machine Learning and Neural Network	3	0	3.0	
CSE	0619 3294	Machine Learning and Neural network Lab	0	3	1.5	
CSE	0714 3273	Data Communication	2	0	2.0	
CSE	0714 3275	Digital Signal Processing	3	0	3.0	0541 1203D, 0541 2204D
CSE	0714 3276	Digital Signal Processing Lab	0	3	1.5	
CSE	0714 3279	Computer Architecture	3	0	3.0	
Total			**11**	**15**	**18.5**	

Fourth Year: 1st Semester

Offering Dept.	Course Code	Course Title	Hours/Week			Cr.	Prereq.
			Th.	Lab	FW		
CSE	0610 41**	Thesis I / Final Year Project I	0	4		2.0	
CSE	0613 4120	Industrial Training/Attachment/Tour	0		6	3	
CSE	0612 4161	Computer and Information Security	3	0		3.0	
CSE	0612 4162	Computer and Information Security Lab	0	3		1.5	
CSE	0612 4165	Computer Networking	3	0		3.0	0613 3271
CSE	0612 4166	Computer Networking Lab	0	3		1.5	
CSE	0613 4142	Software Project Management	2	0		2.0	
CSE	0*** 41**	Option I	3	0		3.0	
CSE	0*** 41**	Option I Lab	0	3		1.5	
Total			**11**	**13**	**6**	**20.5**	

CSE	0610 4150	Final Year Project I	0	4	2.0	
CSE	0610 4180	Thesis I	0	4	2.0	

* FW = Field work.

Fourth Year: 2nd Semester

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.	Prereq.
			Th.	Lab		
CSE	0610 42**	Thesis II / Final Year Project II	0	8	4.0	
CSE	0610 4290	Comprehensive Viva Voce	0	3	1.5	
CSE	0613 4231	Computer Graphics and Multimedia System	3	0	3.0	
CSE	0613 4232	Computer Graphics and Multimedia System Lab	0	3	1.5	
CSE	0688 4247	Engineering Ethics and Cyber Law	2	0	2.0	
CSE	0*** 42**	Option II	3	0	3.0	
CSE	0*** 42**	Option II Lab	0	3	1.5	
Total			**8**	**17**	**16.5**	

CSE	0610 4250	Final Year Project II	0	8	4.0	
CSE	0610 4280	Thesis II	0	8	4.0	

List of Optional Courses

(CSE is the offering department for all optional courses)

Course Code	Course Title	Hours/Week		Cr.	Prerequisite
		Th.	Lab		
0612 4261	Security Engineering	3	0	3.0	
0612 4262	Security Engineering Lab	0	3	1.5	
0612 4163/4263	Advanced Database System	3	0	3.0	
0612 4164/4264	Advanced Database System Lab	0	3	1.5	
0612 4265	Cloud Computing	3	0	3.0	
0612 4266	Cloud Computing Lab	0	3	1.5	
0612 4167/4267	Distributed and Parallel Computing	3	0	3.0	
0612 4168/4268	Distributed and Parallel Computing Lab	0	3	1.5	
0613 4131	Digital Image Processing	3	0	3.0	
0613 4132	Digital Image Processing Lab	0	3	1.5	
0613 4137/4237	Advanced Data Structure and Algorithm	3	0	3.0	0613 1237, 0613 2137
0613 4138/4238	Advanced Data Structure and Algorithm Lab	0	3	1.5	
0613 4143/4243	Natural Language Processing	3	0	3.0	0613 2247
0613 4144/4244	Natural Language Processing Lab	0	3	1.5	
0613 4147/4247	Compiler Construction	3	0	3.0	
0613 4148/4248	Compiler Construction Lab	0	3	1.5	
0619 4193/4293	Deep Learning	3	0	3.0	0542 2102D, 0541 1203D, 0613 3293
0619 4194/4294	Deep Learning Lab	0	3	1.5	
0688 4195/4295	Human Computer Interaction	3	0	3.0	
0688 4196/4296	Human Computer Interaction Lab	0	3	1.5	
0688 4197/4297	Bio-informatics	3	0	3.0	0613 2137
0688 4198/4298	Bio-informatics Lab	0	3	1.5	
0714 4171/4271	Mobile and Wireless Communication	3	0	3.0	0714 3271
0714 4172/4272	Mobile and Wireless Communication Lab	0	3	1.5	
0714 4173/4273	Internet of Things	3	0	3.0	
0714 4174/4274	Internet of Things Lab	0	3	1.5	
0714 4177/4277	VLSI Design	3	0	3.0	0714 2201D
0714 4178/4278	VLSI Design Lab	0	3	1.5	
0611/0612/ 0613/0619 4187	Contemporary Course on Computer Science & Engineering I	3	0	3.0	
0611/0612/ 0613/0619 4188	Lab on Contemporary Course on Computer Science & Engineering I	0	3	1.5	
0611/0612/ 0613/0619 4287	Contemporary Course on Computer Science & Engineering II	3	0	3.0	
0611/0612/ 0613/0619 4288	Lab on Contemporary Course on Computer Science & Engineering II	0	3	1.5	

Summary

Semester	Hours/Week			Credits
	Theory	Lab	Field Work	
Semester-1	14	11		19.5
Semester-2	12	16		19.5
Semester-3	15	09		19.5
Semester-4	13	21		23.5
Semester-5	15	15		22.5
Semester-6	11	15		18.5
Semester-7	11	13	6	20.5
Semester-8	08	17		16.5
Total	**99**	**117**		**160.0**

	Dept.	Hours/Week			Credits
		Theory	Lab	Field work	
Non-Major courses (26.25%)	EEE	9	9		13.5
	MAT	9	0		9.0
	PHY	3	3		4.5
	ENG	2	2		3.0
	STA	3	0		3.0
	BUS	3	0		3.0
	ECO	3	0		3.0
	IPE	0	6		3
	Total	**32**	**20**		**42**
Major course (73.75%)	CSE	**67**	**97**	**6**	**118**
Total		**99**	**117**		**160.0**

Condition to receive degree from the Department of CSE

Total Credit to complete		160
Major courses (from CSE)	Non-core courses	0714 3275 Digital Signal Processing 0714 3276 Digital Signal Processing Lab Option I (Theory and Lab) Option II (Theory and Lab)
	*Core courses	All major courses except mentioned above.
Non-Major courses (from related discipline)	Non-core courses	0715 2108D Computer Aided Drawing
	*Core courses	All non-major courses except mentioned above

***Core courses are mandatory courses for the degree**

COURSE PROFILE
Shahjalal University of Science and Technology
School of Applied Sciences and Technology
Department of Computer Science and Engineering

First Year

Course Title: Structured Programming Language	**Credits**: 3.0
Course Code: 0613 1133	**Contact hours**: 3 hours/week

Rationale:

To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To facilitate students with the necessary knowledge about basic understanding of computer hardware and how a computer works.
- To make the students understand the basic terminology used in computer programming
- Helping the students to develop ability in how to write, compile and debug programs in C language
- Helping the students to develop ability in writing programs involving decision structures, loops, functions, and pointers
- To make the students understand the basic data structures and their implementation
- To enhancing the skill on implementing different searching and sorting techniques
- To enhancing the skill on building up their own logics and implementing them while solving real-world problems

Course Contents:

Basic concept, Overview of programming languages, Problem Solving Techniques, and Data Flow Diagram. C-Language: Preliminaries, Program constructs, variables, and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Control statement, Loops and Nested loops; break, continue, goto, Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays.; Structures within a structure. Automatic, external, static variable, Files; File functions for sequential and Random I/O. Pointers; Pointers and structures, union; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation. Bit field; Advanced features; Preprocessor and Macros, enumeration, Standard library. Recursion: Basic idea of recursion (3 laws-base cases, call itself, move towards the base case by state change), tracing output of a recursive function, applications: factorial, Fibonacci, tower of Hanoi, merge sort, permutation, combination. Sorting: Insertion sort, selection sort, bubble sort. Searching: Linear search, binary search.

Course Learning Outcomes: After the successful completion of the course, the student will be able to

CO 1	Understand the concepts of computer hardware, software and recall the basic terminology used in computer programming
CO 2	Apply control-flow tools such as loop, if-else, etc. pointers, functions, structures, files
CO 3	Construct, compile and debug programs in C language
CO 4	Evaluate real-life problems like sorting, searching etc. using programming terminologies

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3									
CO 3			3									
CO4												3

Textbook:

1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Structured Programming Language Lab	**Credits:** 3.0
Course Code: 0613 1134	**Contact hours:** 6 hours/week

Rationale:
To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To facilitate students with the necessary knowledge about basic understanding of computer hardware and how a computer works.
- To make the students understand the basic terminology used in computer programming
- Helping the students to develop ability in how to write, compile and debug programs in C language
- Helping the students to develop ability in writing programs involving decision structures, loops, functions, and pointers
- To make the students understand the basic data structures and their implementation
- To enhancing the skill on implementing different searching and sorting techniques
- To enhancing the skill on building up their own logics and implementing them while solving real-world problems

Course Contents:
Programming Language: Basic concept, Overview of programming languages, Problem Solving Techniques and Data Flow-Diagram. **Language:** Preliminaries, Program constructs, variables, and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Control statement, Loops and Nested loops; break, continue, go to, Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Automatic, external, static variable, Files; File functions for sequential and Random I/O. Pointers; Pointers and structures, union; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Preprocessor and Macros, enumeration, Standard library. **Recursion:** Basic idea of recursion (3 laws-base cases, call itself, move towards the base case by state change), tracing output of a recursive function, applications: factorial, Fibonacci, tower of Hanoi, merge sort, permutation, combination. **Sorting:** Insertion sort, selection sort, bubble sort, merge sort, quick sort, distribution sort (counting sort, radix sort, bucket sort). **Searching:** Linear search, binary search, application of Binary Search- finding an element in a sorted array, finding nth root of a real number, solving equations. **Stack and Queue**: Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to postfix expressions conversion and evaluation, balancing parentheses using stack, basic queue operations (enqueue, dequeue), circular queue/dequeue, queue-class implementation using array and linked list, application- Josephus problem, palindrome checker using stack and queue.

Course Learning Outcomes: After the successful completion of the course, the student will be able to

CO 1	Understand the concepts of computer hardware, software and recall the basic terminology used in computer programming
CO 2	Apply control-flow tools such as loop, if-else, etc. pointers, functions, structures, files
CO 3	Construct, compile and debug programs in C language
CO 4	Evaluate real-life problems like sorting, searching etc. using programming terminologies

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3									
CO 3			3									
CO4												3

Textbook

1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Discrete Mathematics	**Credits:** 3.0
Course Code: 0541 1143	**Contact hours:** 3 hours/week

Rationale:
This course is designed to introduce first-year CSE students to the fundamental concepts of discrete mathematics. Through this course, they will gain familiarity with mathematical ideas of relevance to engineering on a rigorous footing. Along the way, they are expected to achieve proficiency in logical reasoning and analytical thinking.

Objectives:
- Help them conceptualize basic theories in mathematical reasoning and appreciate the precision of language and rigor required for mathematics.
- Help them conceptualize basic theories in combinatorial analysis to be able to solve counting problems.
- To facilitate necessary knowledge about how to work with discrete data structures like graphs and trees.
- To facilitate necessary knowledge about algorithmic techniques and to be able to implement them in computer programs.
- Apply the knowledge of discrete mathematics in real-life problems using modeling.

Course Contents:
Set, Relations, Functions: Set, Function, Representing Relations, Equivalence Relations. **Propositional Calculus:** Propositions, Predicate and Quantifier. **Algorithms:** Complexity, Divisions, Algorithm, Application of Number Theory. **Recursion:** Sequences and summations, Recursive Definition and algorithm. **Combinatorial Analysis:** Permutation and Combination, Divide and Conquer Algorithms, Generating Functions. **Graphs:** Representation, Isomorphism, Connectivity, Euler and Hamilton path, Shortest path, Planer, Coloring. **Trees:** Spanning trees, Rooted Trees, Binary Trees, Huffman Trees. **Boolean Algebra:** Number System, Boolean Function, representing Boolean Function, Logic gate, Minimization of Circuits.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Apply principles of sets, relations, logic, and Boolean algebra to model discrete structures and simplify digital logic circuits.
CO 2	Analyze computational problems using algorithmic thinking, number theory, and recursive techniques.
CO 3	Solve counting problems using combinatorial methods and apply Boolean algebra to digital logic design
CO 4	Analyze and optimize real-world problems using graph and tree structures, including pathfinding and data encoding.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3									
CO 3		3										
CO 4				3								

Textbook
1. Discrete Mathematics and Its Applications by Kenneth H. Rosen

Course Title: Capstone Project Work I	**Credits:** 1.5
Course Code: 0610 1250	**Contact hours:** 3 hours/week

Rationale:
This is a project which enables the freshmen to apply their novel acquired knowledge to some of the basic real world problem solving.

Objectives:
- Helping the students to develop ability in real life problem solving
- To enhance skill on problem solving
- To help them apply the knowledge of programming, data structure and algorithm

Course Contents:
Any project based on C language including implementation of Data Structure is acceptable. Gaming project using the graphics.h library in C is preferable.

Teachers must have to ensure every project is unique. Innovative project ideas should get extra weight to prevent imitating old projects.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Apply programming knowledge to create visible products
CO2	Enhance problem-solving capability by designing logical structures to divide and solve problems with scientific and technical knowledge
CO3	Habituate to work as an efficient team member
CO4	Present ideas and projects in front of audience

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3		2		2							3
CO 2	3	3		2								
CO 3									3		2	
CO 4										3	2	

Course Title: Data Structure	**Credits:** 3.0
Course Code: 0613 1237	**Contact hours:** 3 hours/week

Rationale:

To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To explain the purpose and mathematical background of algorithm analysis
- To facilitate necessary knowledge about the abstract data types,such as, stacks, queues and deques

- To familiarize with variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated
- To facilitate necessary knowledge about the characteristics and optimal behavior of hash tables for access and retrieval
- To provide the knowledge of various sorting algorithms and the run-time analysis required to determine their efficiencies
- To help them understand various tree traversal techniques and graph algorithms

Course Contents:

Internal Data Representation: Specification, representation, Asymptotic analysis: Recurrences, Substitution method and manipulation of basic data structures: arrays, records and pointers, linked lists, stacks, queues, recursion, trees, optimal search trees, heaps, disjoint sets. **Recursion:** permutation, combination. **Sorting:** merge sort, quick sort (randomized quick sort), distribution sort (counting sort, radix sort, bucket sort), lower bounds for sorting, external sort. **Binary Tree:** Binary tree representation using array and pointers, traversal of Binary Tree (in-order, pre-order and post-order). **Ternary tree, Binary Search Tree:** BST representation, basic operations on BST (creation, insertion, deletion, querying and traversing), application- searching, sets. Ternary search tree, Binary Index tree, Segment tree, RMQ (Range Minimum Query). **Searching: A**pplication of Binary Search- finding element in a sorted array, finding nth root of a real number, solving equations. **Heap:** Min-heap, max-heap, Fibonacci-heap, applications-priority queue, heap sort. **Set Operations& Disjoint Set:** Set representation using bitmask, set/clear bit, querying the status of a bit, toggling bit values, LSB, application of set operations, union find, path compression. **Huffman Coding Graph:** Graph representation (adjacency matrix/adjacency list), basic operations on graph (node/edge insertion and deletion), **Traversing a graph:**Review of Breadth first search (BFS), Depth first search (DFS), Topological Sort, Strongly Connected Components, Euler Path, Articulation Point, Bridge, Bi-connected Components, graph-bicoloring, Floodfill, Dijkstra's Shortest Path Algorithm, Bellman –Ford algorithm and negative cycle detection, Floyd-Warshall all pair shortest path algorithm, Johnson's algorithm, shortest path in Directed Acyclic Graph. **Minimum spanning tree:** Prim's algorithm and Kruskal's algorithm. **Self-Balancing Binary Search Tree:** AVL tree (rotation, insertion). **String ADT:** The concatenation of two strings, the extraction of substrings, searching a string for a matching substring, parsing, Suffix tree, Suffix array.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Define and explain the fundamental data structures such as lists, queues, trees, etc.
CO 2	Illustrate the concept of algorithm complexity analysis
CO 3	Compare tradeoffs in the design and implementations of the data structures
CO 4	Select appropriate algorithms to use in specific applications and apply including real life problem

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Textbook

1. Advanced Data Structures, Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms Thomas H. Cormen , Charles E. Leiserson

Course Title: Data Structure Lab	**Credits:** 1.5
Course Code: 0613 1238	**Contact hours:** 4 hours/week

Rationale:

To provide the students with solid foundations for implementation of various linear and non-linear data structures. To teach the students how to select, design and develop data structures and algorithms that are appropriate for problems that they might encounter.

Objectives:

- To facilitate necessary hands-on knowledge to implement and manipulate various linear and non-linear data structures
- To facilitate necessary hands-on knowledge to implement various searching and sorting algorithms
- To facilitate necessary hands-on knowledge to design and develop real-world applications using suitable data structure.

Course Contents:

This course is based on the theory course 0613-1237. This course has been design to facilitate students a hands-on experience to implement various types of linear and non-linear data structures and related algorithms to manipulate those data structures. Besides, students will implement various sorting and searching algorithms also. At the end of the course the students should be able to design and develop data structures and sorting and searching algorithms that are used in various real-world applications.

Here are the data structures and algorithms that will be implemented by the students.

- Binary and linear searching using arrays; representing stack and queue using arrays
- Different types of linked lists; representing stacks and queues using linked list
- Converting infix expression to postfix and evaluating postfix expressions
- Implementing few algorithms using recursion
- Implementing various sorting algorithms
- Creating, manipulating and traversing Binary and Ternary trees
- Creating, manipulating, traversing and searching BSTs
- Creating and manipulating max/min heaps and implementing merge sort
- Implementing Kruskal's minimum spanning tree algorithm using disjoint sets
- Implementing compression algorithm using Huffman Coding graph
- Creating, manipulating, traversing and searching various types of graphs
- Implementing few well-known graph based algorithms
- Implementing, Prim's algorithm for finding minimum spanning tree
- Creating, manipulating, traversing AVL trees; solving few string based problems using string operations.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Understand various data representation techniques in the real world.
CO 2	Implement linear and non-linear data structures.
CO 3	Analyze various algorithms based on their time and space complexity
CO 4	Design and develop real-world applications using suitable data structure.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12

CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Textbook

1. Advanced Data Structures, Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms Thomas H. Cormen , Charles E. Leiserson

Second Year

Course Title: Object Oriented Programming Language	**Credits:** 3.0
Course Code: 0613 2133	**Contact hours:** 3 hours/week

Rationale:

Students wishing to build up their career in CSE need to develop software to solve problems and this course will help them learn the basics of OOP and OOP programming using JAVA.

Objectives:

- To help students conceptualize basic theories and principles of object-oriented programming;
- Helping the students to develop the ability in applying the concepts of data encapsulation, inheritance, and polymorphism to large-scale software
- To facilitate necessary knowledge about good programming practices and how to write modular codes with the help of OOP concepts.
- To provide knowledge of packages, and how to work with them. Also, give students training to code reusable programs with JAVA.
- To make students understand how to work with JAVA generic templates to design Classes and data structures that can work with different data types.

Course Contents:

Introduction to Java: History of Java**,** Java Class Libraries**,** Introduction to Java Programming**,** A simple Program, The Bytecode**. Data Types, Variables, and Arrays:** Strongly Typed Language, The Primitive Types, Literals, Variables, Type Conversion and Casting, Arrays, Strings. **Operators:** Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment & ? Operator, Operator Precedence. **Control Statements:** Java's Selection Statements: if & switch, Iteration Statements: while, do-while, for, for-each & nested loops, Jump Statements: break, continue, return**. Introducing Classes:** Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection. **A Closer Look at Methods and Classes:** Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, final, Nested and Inner Classes, Command-Line Arguments & Variable-Length Arguments. **Inheritance:** Inheritance Basics, Using super, Creating a Multilevel Hierarchy, Method Overriding, Dynamic Method Dispatch, Abstract Classes, final with Inheritance, Local Variable Type Inference and Inheritance, The Object Class**. Packages and Interfaces:** Packages and Member Access, Importing Packages, Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods. **Exception Handling:** Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws & finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions. **Multithreaded Programming:** The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive and join, Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State, Using Multithreading. **Enumerations, I/O, Generics:** Enumeration Fundamentals, Type Wrappers, I/O Basics, Console Input/Output, The I/O Classes and Interfaces, I/O Exceptions, Stream Classes, Serialization, A Simple Generics Example, The General Form of a Generic Class, Bounded Types, Wildcard Arguments, Generic Interfaces, Generic Class Hierarchies. **Lambda Expressions, Modules, Records:** Introducing Lambda Expressions, Block Lambda Expressions, Passing Lambda Expressions as Arguments, Module Basics, Legacy Code and the Unnamed Module, Exporting to a Specific Module, Module Graphs, Record Basics, Text Blocks, Sealed Classes & Interfaces. **String Handling, Collections Framework, Utility Classes:** The String Constructors, String Operations, String Methods, StringBuffer, StringBuilder, Collection Interfaces, Collection Classes, Spliterators, Working with Maps, The Collection Algorithms, StringTokenizer, Date, Formatter, Scanner. **Event Handling, AWT, Swing:** Event Handling Mechanisms, Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter Classes, AWT Classes, Window Fundamentals, AWT Control Fundamentals, The Origins of Swing, Swing Features, Components and Containers, Swing Packages, Painting in Swing, Swing Menus.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Recognize the basic syntax, compilation, and execution order and process of Java programming language.
CO 2	Describe key concepts of the object-oriented programming paradigm.
CO 3	Interpret real-world problems in terms of objects rather than procedures.
CO 4	Apply object-oriented programming principles to implement small and medium-scale Java programs with simple graphical user interfaces.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2	3											
CO 3				3								3
CO 4		3										

Textbook

1. Java: The Complete Reference by Herbert Schildt.
2. Introduction to Programming in Java, Robert Sedgewick & Kevin Wayne.
3. An Introduction to Object-Oriented Programming, Timothy Budd.
4. Java-How to Program by Deitel & Deitel.

Course Title: Object Oriented Programming Language Lab	**Credits:** 1.5
Course Code: 0613 2134	**Contact hours: 3** hours/week

Rationale:

Students wishing to build up their career in CSE need to develop software to solve problems and this course will help them learn how to design and

implement small to medium-scale Java programs with simple graphical user interfaces applying Object Oriented Programming principles.

Objectives:

- To help students develop the ability to write programs using Java programming language
- To help students develop the ability to design and develop software using theories and principles of object-oriented programming
- To help students develop the ability to interpret real-world problems in terms of objects rather than procedures.
- To help students achieve the ability to develop software as a team member and effectively communicate between team members.
- To make students able to lead and manage a software development project

Course Contents:

Introduction to Java: Introducing Java development environment setup, Compiling and Executing A simple Program**. Data Types, Variables, and Arrays:** Strongly Typed Language, The Primitive Types, Literals, Variables, Type Conversion and Casting, Arrays, Strings. **Operators:** Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment & ? Operator, Operator Precedence. **Control Statements:** Java's Selection Statements: if & switch, Iteration Statements: while, do-while, for, for-each & nested loops, Jump Statements: break, continue, return**. Introducing Classes:** Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection. **A Closer Look at Methods and Classes:** Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, final, Nested and Inner Classes, Command-Line Arguments & Variable-Length Arguments. **Inheritance:** Inheritance Basics, Using super, Creating a Multilevel Hierarchy, Method Overriding, Dynamic Method Dispatch, Abstract Classes, final with Inheritance, Local Variable Type Inference and Inheritance, The Object Class**. Packages and Interfaces:** Packages and Member Access, Importing Packages, Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods. **Exception Handling:** Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws & finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions. **Multithreaded Programming:** The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive and join, Thread Priorities, Synchronization, Interthread

Communication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State, Using Multithreading. **Enumerations, I/O, Generics:** Enumeration Fundamentals, Type Wrappers, I/O Basics, Console Input/Output, The I/O Classes and Interfaces, I/O Exceptions, Stream Classes, Serialization, A Simple Generics Example, The General Form of a Generic Class, Bounded Types, Wildcard Arguments, Generic Interfaces, Generic Class Hierarchies. **Lambda Expressions, Modules, Records:** Introducing Lambda Expressions, Block Lambda Expressions, Passing Lambda Expressions as Arguments, Module Basics, Legacy Code and the Unnamed Module, Exporting to a Specific Module, Module Graphs, Record Basics, Text Blocks, Sealed Classes & Interfaces. **String Handling, Collections Framework, Utility Classes:** The String Constructors, String Operations, String Methods, StringBuffer, StringBuilder, Collection Interfaces, Collection Classes, Spliterators, Working with Maps, The Collection Algorithms, StringTokenizer, Date, Formatter, Scanner. **Event Handling, AWT, Swing:** Event Handling Mechanisms, Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter Classes, AWT Classes, Window Fundamentals, AWT Control Fundamentals, The Origins of Swing, Swing Features, Components and Containers, Swing Packages, Painting in Swing, Swing Menus.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Write and execute a basic java program using proper syntax, compilation, and execution procedure of Java.
CO 2	Model classes from real-world problems in terms of objects rather than procedures.
CO 3	Apply object-oriented programming principles to implement small and medium-scale Java programs with simple graphical user interfaces.
CO 4	Design Java programs for complex problems, making good use of the features of the language such as classes, inheritance, polymorphism, abstraction, package, and interface.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Java: The Complete Reference by Herbert Schildt.

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									3
CO 4				3								

2. Introduction to Programming in Java, Robert Sedgewick & Kevi Wayne.
3. An Introduction to Object-Oriented Programming, Timothy Budd.
4. Java-How to Program by Deitel & Deitel.

Course Title: Algorithm Design and Analysis		**Credits:** 3.0
Course Code: 0613 2137	**Contact hours:** 3 hours/week	

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:
- To familiarize with the asymptotic performance of algorithms
- To familiarize with rigorous correctness proofs for algorithms
- To demonstrate a familiarity with major algorithms and data structures
- To facilitate with necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Contents:
Analysis of Algorithm: Asymptotic analysis: Recurrences, Substitution method, Recurrence tree method, Master method **Hash Table:** Hash tables, hash function, open addressing, perfect hashing, single and multi probehasing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, job scheduling with deadline. **Dynamic Programming:** Elements of DP (Optimal substructure, Overlapping sub problem), Coin change related problem, 0-1 knapsack, Longest Common Subsequence finding problem, LCS and LIS/LDS variations, Matrix Chain

Multiplication.**Red black Tree and Binomial Heaps, Stassen's algorithm Network Flow:** Flow Networks, Max-Flow Min-cut theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum bipartite matching, minimum path cover, edge cover. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. For example in traveling salesman problems. **Geometric algorithm:** Segment-segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. **RSA public key generation,** NP Completeness, NP hard and NPcomplete problems. **String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree and Suffix Array. Basic combinatorics, Probability and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT and FFT

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Analyze the computational and memory complexities of algorithms
CO 2	Prove the correctness of algorithms
CO 3	Explain how and why the algorithms work
CO 4	Apply the algorithms to solve real life problems

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Textbook

1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: Algorithm Design and Analysis Lab	**Credits:** 1.5
Course Code: 0613 2138	**Contact hours: 3** hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To familiarize with the asymptotic performance of algorithms.
- To familiarize with rigorous correctness proofs for algorithms.
- To demonstrate a familiarity with major algorithms and data structures.
- To facilitate with necessary knowledge about important algorithmic design paradigms and methods of analysis.
- To develop skills to synthesize efficient algorithms in common engineering design situations.

Course Contents:
Hash Table: Hash tables, hash function, open addressing, perfect hashing, single and multi probehasing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, job scheduling with deadline. **Dynamic Programming:** Elements of DP (Optimal substructure, Overlapping sub problem), Coin change related problem, 0-1 knapsack, Longest Common Subsequence finding problem, LCS and LIS/LDS variations, Matrix Chain Multiplication. **Red black Tree and Binomial Heaps, Stassen's algorithm Network Flow:** Flow Networks, Max-Flow Min-cut theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum bipartite matching, minimum path cover, edge cover. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. For example in traveling salesman problems. **Geometric algorithm:** Segment-segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. **RSA public key generation,** NP Completeness, NP hard and NPcomplete problems.**String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree and Suffix Array. Basic combinatorics, Probability and Game

theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT and FFT.

Implement all the Contents related to the coursework 0613 2137.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Implement all the listed algorithms.
CO 2	Construct bug free and efficient codes for the algorithms.
CO 3	Construct efficient solutions of complex problems using suitable algorithms
CO 4	Analyze the computational and memory complexities of algorithms.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Textbook

1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: Capstone Project Work II	**Credits:** 3
Course Code: 0610 2250	**Contact hours:** 6 hours/week

Rationale:

This course is based on project work. Target of this course is to involve students in real life software development which will help to increase their skill to reach the requirements of the software industry. Also this course will help students to improve their communication skill and to present their work in front of an audience.

Objectives:

- To facilitate necessary knowledge about latest technology
- To develop skills on software development
- To develop skills on teamwork and presentation

Course Contents:

Project focusing on an Object oriented programming approach and using standard algorithms is preferable. Every project should maintain a goal so that it can be used as a useful tool in the IT fields. Also innovative project ideas that require different types of scripting/programming languages or programming tools can be accepted with respect to the consent of the corresponding project supervisor.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Apply latest state of the art technologies
CO 2	Design and implement ideas for complete software
CO 3	Evaluate existing computer and mobile applications
CO 4	Explain ideas to groups and present their noble findings

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Course Title: Web Technologies	**Credits:** 1.5
Course Code: 0612 2260	**Contact hours:** 3 hours/week

Rationale:

Websites and web applications behold a very large portion of the software industry. Technologies related to this sector are getting updated very frequently. To keep pace with this continuously upgrading world students must have a very good rudimentary level knowledge of web technologies.

Objective of this course is to make students introduced to this sector and prepare them for the industry by providing them necessary hands-on tools.

Objectives:

- To facilitate with rudimentary level knowledge of web technologies
- To provide knowledge on development of web applications
- To make students understand server-side, front-end technologies
- To make students understand Model View Controller model
- To provide knowledge on plug-ins, APIs, extensions

Course Contents:

Concepts of Web Engineering, Requirements Engineering and Modeling Web Applications, Web Application Architectures, Technologies and Tools for Web Applications, Testing and Maintenance of Web Applications, Usability and Performance of Web Applications, Security of Web Applications, The Semantic Web, design methods and technologies, interface design, usability of web applications, accessibility, testing, metrics, operation and maintenance of Web applications, security, and project management, client-side (XHTML, JavaScript, and CSS) and server-side (Perl and PHP) architecture, Web engineering concepts behind the frameworks of Joomla, Drupal, Wordpress. **Server-side technology:** LAMP, Web application frameworks (example: Silverlight, Adobe Flex), Web 2.0 and Web APIs. **Front-end technology:** HTML, XHTML, XML. CSS styling, layout, selector, Document object model and JavaScript. **Client-Programming:** Web APIs with JavaScript (example: Google Ajax API). **MVC:** Understanding Model, view and controller Model. **Understanding Web APIs:** REST, XML, JSON, RSS Parsing. **JavaScript Exercise:** The goal of this assignment is to allow you to explore and use as many of JavaScript's objects, methods, and properties as possible in a small assignment. Some functions must be written from scratch. Other functions, appropriately attributed, may be downloaded from the web and used as a part of the system or as the basis for your own functions. **PHP Exercise:** Build a set of PHP scripts that perform some dynamic server side functionality. **Understanding plug-ins:** Develop a Firefox extension.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Describe fundamental knowledge of web technologies
CO 2	Design web applications maintaining necessary criteria
CO 3	Implement their knowledge of server-side technologies
CO 4	Implement their knowledge of front end technologies
CO 5	Design web applications maintaining MVC framework
CO 6	Develop plug-ins, APIs, extensions
CO 7	Use Javascript, PHP and other contemporary technologies to develop web applications

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								
CO 5				3								1
CO 6										3		
CO 7											2	

Course Title: Introduction To Competitive Programming	**Credits:** 1.5
Course Code: 0613 2234	**Contact hours:** 3 hours/week

Rationale:

This course is designed to help both beginners and intermediate programmers alike to dominate the algorithms and data structures necessary to do well in programming contests and to gain a competitive edge over candidates in software interviews. Techniques and applications that are useful for the field, focusing on real problems and how they are solved are expected to be covered in this course.

Objectives:

- To facilitate the necessary knowledge about advanced data structures and algorithms
- To enhance the skill of problem-solving
- To get trained with sprint and marathon contests
- To help to improve thinking capability and solution formulation

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3										
CO 2								3				
CO 3										3		
CO 4												3

Course Contents:

Data Structure: Trie Tree, BIT, Segment Tree, Splay Tree, MO's Algorithm, Square Root Decomposition, Heavy Light Decomposition, Persistent Data Structure (Segment Tree, Trie), DSU on Tree, Treap, K-D Tree, KNN Tree, Sparse Table.

String Processing: KMP, Suffix Array, Suffix Automata, Suffix Tree, Palindromic Tree, Aho-Corasick, Manacher Algorithm, Extended KMP, Hashing (Rolling Hash).

Game Theory: Nim Game, Sprague-Grundy Value, Green Hackenbush, Blue Red Hackenbush, Blue Red Green Hackenbush, Colon Principle, Fusion Principle.

Combinatorics & Probability: Burnside Lemma, Inclusion Exclusion, Combination, Permutation, Catalan Number, Stirling Number, Probability, Expected Value.

Number Theory: Chinese Remainder Theorem, Euler Phi, Extended Euclid, Prime Factorization, Mobius Function, Primitive Prime, Huge Mod.

Basic Math: FFT, DFT, NTT, Gaussian Elimination, Matrix Exponentiation.

Basic Geometry: Fundamental Concepts of Geometry, Closest Pair of Point, Convex Hull, Rectangle Union, Circle Union, Polygon Clipping, Line Sweep, Line Intersection.

Dynamic Programming: Coin change variations, knapsack, LIS, LCS, Edit Distance, etc.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Apply the data structures and algorithms to design solutions to critical programming problems
CO 2	Avoid doing plagiarism in the contest time regarding it as a crime
CO 3	Explain the idea, used algorithm and data structure behind to solve a critical problem with others
CO 4	Prepare themselves to deal with competitive environments by attending frequent problem solving contests

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Competitive Programming 3 by Steven Halim.
2. 102 Combinatorial Problems by Titu Andreescu & Zuming Feng.
3. Problem-Solving Methods in Combinatoricsby Pablo Soberón.
4. Art of Programming Contest by Ahmed ShamsulArefin.
5. Programming Challenges: The Programming Contest Training Manual bySteven S Skiena, Miguel A. Revilla.

Course Title: Theory of Computation	**Credits:** 3.0
Course Code: 0613 2247	**Contact hours:** 3 hours/week

Rationale:

Theory of computation lays a strong foundation for a lot of abstract areas of computer science. It is used in Artificial Intelligence, Natural Language Processing, Probability or Computer vision, in certain areas of mathematics like Number theory. So Students wishing to build up their career in such a field of CSE need to achieve a better understanding of this topic.

Objectives:

- To familiarize with the formalization of the notion of problems via formal languages
- To familiarize with the formalization of the notion of computation using "abstract computing devices" called automata
- To help them understanding a hierarchy of classes of problems or formal languages (regular, context-free, context-sensitive)
- To help them understanding a hierarchy of classes of automata (finite automata, pushdown automata, and Turing machines)
- To facilitate with the knowledge about complexity classes P and NP, and Intractability (NP-completeness)
- To facilitate with the knowledge about space complexity: NL-completeness and PSPACE-completeness

Course Contents:

Introduction: Automata Theory ,Mathematical Notions and Terminology , Sets , Strings and languages , Definitions, Theorems, and Proofs, Types of Proof - Proof by construction - Proof by contradiction. **Finite Automata & Regular Languages:** Finite Automata - Formal definition of a finite automaton - Examples of finite automata - Formal definition of computation - Designing finite automata - The regular operations ,State minimization of finite Automata, Nondeterminism - Formal definition of a nondeterministic finite automaton - Equivalence of NFAs and DFAs - Closure under the regular operations Regular Expressions - Formal definition of a regular expression - Equivalence with finite automata, GTG, Non-regular-languages, The pumping lemma for regular languages. **Context-Free Languages**: Context-Free Grammars - Formal definition of a context-free grammar - Examples of context-free grammars - Designing context-free grammars - Ambiguity Normal Form: Chomsky normal form (CNF), GNF, and Simple grammar. **Pushdown Automata** - Formal definition of a pushdown automaton - Examples of pushdown automata - Equivalence with context-free grammars Deterministic Context-Free Languages - Properties of DCFLs - Deterministic context-free grammars, Relationship of DPDAs and DCFG, Relationship of DPDAs and DCFG. **The Church–Turing Thesis**: Turing Machines - Formal definition of a Turing machine - Examples of Turing machines Variants of Turing Machines - Multitape Turing machines - Nondeterministic Turing machines - Enumerators - Equivalence with other models

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Describe the fundamental elements and branches of theory of computation and its features
CO 2	Design grammar for a language
CO 3	Design regular expression to generalize all the elements of a language
CO 4	Design deterministic and nondeterministic state diagrams to find out acceptable and non-acceptable elements for a specific language

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Textbook

1. An Introduction to Formal Languages and Automata. Sixth Edition, Peter Linz
2. Introduction to the Theory of Computation by Michael Sipser, (Latest Edition).
3. Introduction to Languages and the Theory of Computation, by J. C. Martin.

Course Title: Data Science	**Credits:** 2.0
Course No.: 0688 2291	**Contact hours:** 2 hours / week

Rationale:

Data Science is a rapidly evolving field that studies how to analyze and organize relevant data through appropriate data visualizations. The technical foundation of Data science arises from Mathematics, Statistics and Computer Science. Those with a technical background related to data science need an understanding of the data relevant to the particular problem application area. Those with expertise in the application area must acquire the relevant technical knowledge in order to effectively and accurately make use of data science tools and methodologies. This course will build the technical and analytical skills required to collect, clean, and model data and show a path to bring all of these skills together in the creation and presentation of a data analytics predictive model, software system, or visualization.

Objectives:

- To introduce the fundamentals of data analytics and data science.
- To facilitate knowledge about data visualizations and appropriate analysis.
- To acquaint students with the methods to store and access data from a variety of sources.
- To familiarize with techniques and tools for transformation of Data.
- To help to accumulate basic ideas about statistical methods, regression techniques, and machine learning algorithms to make sense out of data sets both large and small.

Course Contents:

Introduction to Data Science, The scope of Data Science, Descriptive Statistics and Exploratory Data Analysis. Data Scraping, Cleaning and Summarization. Statistical Significance and P-values. Principles of Visualizing Data. Building Models and Validating Models. Linear Algebra Review. Linear Regression and Logistic Regression. Large-scale Clustering. Mining Massive Datasets. Crowdsourcing and Ensemble Learning.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Explain Data Science and Big Data and their impact on real life situations
CO 2	Outline procedures for collecting, cleaning and preparing data from necessary domain
CO 3	Apply different methods to summarize and analyze data
CO 4	Create visual representation of data, that has been abstracted in some schematic form, including attributes or variables for the units of information

Mapping of Course Learning Outcomes to Program Learning Outcomes

CO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CO1	3								
CO2		3							
CO3			3						
CO4					3				

Textbook

1. The Signal and the Noise: Why so many predictions fail but some don't, by Nate Silver, Penguin Press.
2. The Art of Data Science, by Roger D. Peng and Elizabeth Matsui.

Course Title: Data Science Lab	**Credits:** 1.5
Course Code: 0688 2292	**Contact hours:** 3 hours/week

Rationale:

Data Science is a rapidly evolving field that studies how to analyze and organize relevant data through appropriate data visualizations. The technical foundation of Data science arises from Mathematics, Statistics and Computer Science. Those with a technical background related to data science need an understanding of the data relevant to the particular problem application area. Those with expertise in the application area must acquire the relevant technical knowledge in order to effectively and accurately make use of data science tools and methodologies. This course will build the technical and analytical skills required to collect, clean, and model data and show a path to bring all of these skills together in the creation and presentation of a data analytics predictive model, software system, or visualization.

Objectives:

- To facilitate necessary knowledge about functionality of Data Science
- To demonstrate how Data Science solves different real-world problems
- To help solve different Data Science problems using appropriate Data visualization, organization and presentation
- To help to develop systems by assembling different solution techniques

Course Contents:

Introduction to Data Science, The scope of Data Science, Descriptive Statistics and Exploratory Data Analysis. Data Scraping, Cleaning and Summarization. Statistical Significance and P-values. Principles of Visualizing Data. Building Models and Validating Models. Linear Algebra Review. Linear Regression and Logistic Regression. Large-scale Clustering. Mining Massive Datasets. Crowdsourcing and Ensemble Learning. Python for data analysis. Data Wrangling with Python, including tools/libraries as Pandas, NumPy, and IPython. Machine Learning using Python.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Analyze and visualize large dataset.
CO 2	Solve problems like water-jug problem, the missionaries and cannibal problem
CO 3	Implement different tools and libraries (Pandas, NumPy, and IPython etc.) provided by Python.
CO 4	Implement different tools and libraries provided by Python for machine learning.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2				3								
CO 3				3								1
CO 4										3		

Textbook

1. The Signal and the Noise: Why so many predictions fail but some don't, by Nate Silver, Penguin Press.
2. The Art of Data Science, by Roger D. Peng and Elizabeth Matsui.

Course Title: Numerical Analysis	**Credits:** 2.0
Course Code: 0541 2257	**Contact hours:** 2 hours/week

Rationale:

There are many interesting or economically pressing problems that do not have the "closed form algebraic solutions". Numerical methods are the answer to that.

Numerical analysis is the story of how functions, derivatives, integrals, and differential equations are handled as strings of numbers in the computer. Many of these problems are too large or too difficult to solve in a conventional manner, for which we resort to using the computer to do the hard work for us. It is intended to introduce the student to the algorithms and techniques an engineer might employ in solving these difficult problems. This course helps us to know how fast errors cause problems and to find better algorithms that cause less error. Therefore, this course is indispensable for all students in almost all disciplines.

Objectives:

- To facilitate necessary knowledge about deriving appropriate numerical methods to solve algebraic and transcendental equations
- Acquaint students with the basic tools to develop appropriate numerical methods to approximate a function, solve a differential equation
- Help them conceptualize basic theories to derive appropriate numerical methods to evaluate a derivative at a value
- Helping the students to develop ability in deriving appropriate numerical methods to solve a linear system of equations
- Make the students understand error analysis mechanism for various numerical methods
- To accumulate basic ideas about various numerical root finding methods
- Foster the analytical and critical ability of the students to derive appropriate numerical methods to calculate a definite integral

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			3								
CO 2	3	3	3	3								
CO 3	3	3		3								

- To make them understand various numerical methods in a modern computer language like Matlab, Python

Course Contents:

Approximation and round off error: Errors in numerical calculations. Error: Definitions, sources, examples. Propagation of Error. A general error formula. Taylor series and reminders. **Root finding:** The bisection method and the iteration method, the method of false position. Newton-raphson method. Roots of polynomials. **Solution of systems of Linear equations:** Gaussian elimination. The pivoting strategy, Iteration method solution of tridiagonal systems. LU decomposition, matrix inverse. **Numerical solution of ordinary differential equations:** Euler's method (including modified form), Runge-Kutta method. **Numerical Integration:**Trapezoidal method. Simpson's method. Weddle's method; Eigen value problems for matrices, Use of computer to implement projects in numerical methods.

Methods of approximation theory: Polynomial interpolation: Lagrange form, divided formula for interpolation. **Regression:** Background, linear regression, non-linear regression. **Optimization**: optimization in one dimension; unconstrained optimization; nonlinear least squares; constrained optimization; iterative linear solvers – gradient descent, conjugate gradient.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Analyze the sources of errors in mathematical operations on the computer and their effects on using numerical algorithms
CO 2	Perform numerical analysis to obtain approximate solutions for various mathematical operations and tasks, such as finding the solution of linear and nonlinear equations, root finding, optimization, interpolation, integration, and the solution of ordinary differential equations
CO 3	Analyze the behavior of various numerical methods and to be able to discuss their stability, their order of convergence and their conditions of application

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Numerical Methods for Engineers – Steven C. Chapra, Raymond P. Canale
2. Numerical Analysis – R.L. Burden, J.D. Faires
3. Scientific Computing: An introductory survey – Michael T. Heath

Course Title: Numerical Analysis Lab	**Credits:** 1.5
Course Code: 0541 2258	**Contact hours:** 3 hours/week

Rationale:

This course introduces students to numerical methods for the solution of basic mathematical problems that cannot be solved by hand. The course aims to introduce students to the toolbox of widely-used numerical methods in computational science. Students will be able to apply these methods to problems in a variety of sciences. It is designed for practical implementations of common algorithms taught in the corresponding theory course – CSE 239. Students will design and implement algorithms directly from pseudo codes, as well as from problem statements new to them.

Objectives:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	3		3							
CO 2	3	3	3	3	3							
CO 3	3	3	3	3								
CO 4								3	3	3		

- To familiarize with the numerical methods used in computational science
- To help to develop skills to apply numerical methods to problems in practice.
- To familiarize with, use, and understand software which uses numerical methods
- To facilitate with the knowledge about the role of numerical methods in science
- To provide basic knowledge of coding various numerical methods in a modern computer language like Matlab, Python

Course Contents:

Approximation and round off error: Measuring error, relative approximate error, relative true error. **Root finding:** Bisection method, false position methods, Newton-raphson method. **Solution of systems of Linear equations:** Naïve Gaussian elimination. Gaussian elimination with partial pivoting, The pivoting strategy, Iteration method solution of tridiagonal systems. LU decomposition, matrix inverse. **Numerical solution of ordinary differential equations:** Euler's method (including modified form), Runge-Kutta method. **Numerical Integration:** Trapezoidal method. Simpson's method. Weddle's method; Eigen value problems for matrices, Use of computer to implement projects in numerical methods. **Methods of approximation theory:** Polynomial interpolation: Lagrange form, Newton's divided difference formula for interpolation. **Regression:** Background, linear regression, non-linear regression. **Optimization**: optimization in one dimension; unconstrained optimization; nonlinear least squares; constrained optimization; iterative linear solvers – gradient descent, conjugate gradient.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Implement algorithms from a known/given pseudocode for solving numerical analysis problem
CO2	Design and/or implement algorithms directly from a given problem statement (without needing any pseudocode)
CO3	Implement bug-free and efficient codes against all algorithms
CO4	Perform teamwork to solve complex real-world problems and communicate their findings on a written report and/or by oral presentations

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Numerical Methods for Engineers – Steven C. Chapra, Raymond P. Canale
2. Numerical Analysis – R.L. Burden, J.D. Faires
3. Scientific Computing: An introductory survey – Michael T. Heath

Third Year

Course Title: Database System	**Credits:** 3.0
Course Code: 0612 3163	**Contact hours:** 3 hours/week

Rationale:

This course is designed to introduce the basic concepts of database, learn the foundations of database systems, focusing on basics such as the relational algebra and data model, schema normalization, query optimization, and transactions.

Objectives:

- To facilitate knowledge about different issues involved in the design and implementation of a database system.
- To make students understand physical and logical database designs, database modeling, relational, hierarchical and network models
- To provide the knowledge of data manipulation language to query, update, and manage a database

- To help to develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, Client/Server (Database Server), Data Warehousing.
- To help apply the knowledge of design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS

.

Course Contents:

Introduction: Purpose of Database Systems, Data Abstraction, Data Models, Instances and Schemes, Data Independence, Data Definition Language, Data Manipulation Language, Database Manager, Database administrator, Database Users, Overall System Structure, Advantages and Disadvantage of a Database Systems. Data Mining and analysis, Database Architecture, History of Database Systems. **Relationship Entity-Model:** Entities and Entity Sets, Relationships and Relationship Sets, Attributes, Composite and Multivalued Attributes, Mapping Constraints, Keys, Entity-Relationship Diagram, Reducing of E-R Diagram to Tables, Specialization, Generalization, Attribute Inheritance, Aggregation, Alternative E-R Notations, Design of an E-R Database Scheme. **Relational Model:** Structure of Relational Database, Fundamental Relational Algebra Operations, the Tuple Relational Calculus, the Domain Relational Calculus, Modifying the Database. **Relational Commercial Language:** SQL, Basic structure of SQL Queries, Query-by-Example, Nested Sub queries, Complex queries, Integrity Constraints, Authorization, Dynamic SQL, Recursive Queries. **Relational Database Design:** Pitfalls in Relational Database Design, Functional Dependency Theory, Normalization using Functional Dependencies, Normalization using Multivalued Dependencies, Normalization using join Dependencies, Database Design Process. **File and System Structure:** Overall System Structure, Physical Storage Media, File Organization, RAID, Organization of Records into Blocks, Sequential Files, Mapping Relational Data to Files, Data Dictionary Storage, Buffer Management. **Indexing and Hashing:** Basic Concepts, Ordered Indices, B+ -Tree Index Files, B-Tree Index Files, Static and Dynamic Hash Function, Comparison of Indexing and Hashing, Index Definition in SQL, Multiple Key Access. **Concurrency Control:** Schedules, Testing for Serializability, Lock-Based Protocols, Timestamp-Based Protocols, Validation Techniques, Multiple Granularity, Multiversion Schemes, Insert and Delete Operations, Deadlock Handling. **Distributed Database:** Structure of Distributed Databases, Trade-off in Distributing the Database, Design of Distributed Database, Transparency and Autonomy, Distributed Query Processing, Recovery in Distributed Systems, Commit Protocols, Concurrency Control.

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3									
CO 3	3											
CO 4											3	

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Describe the basic concepts and appreciate the applications of database systems
CO2	Illustrate the basics of SQL and construct queries using SQL.
CO3	Be familiar with a commercial relational database system by writing SQL using the system.
CO4	Be familiar with the relational database theory and be able to write relational algebra expressions for queries.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Database System Concept, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fourth edition
2. Files and Databases- An Introduction, Peter D. Smith and G.M. Barnes, Addison- Wesley
3. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, Third edition

Course Title: Database System Lab	**Credits:** 1.5
Course Code: 0612 3164	**Contact hours:** 3 hours/week

Rationale:

Database System LAB course will concentrate on the design and implementation of a database system and applying SQL query.

Objectives:

- To introduce the basic concepts of database.
- Developing a real world database application.
- To learn the design of a database starting from the conceptual design
- To the implementation of database schemas and user interfaces to a database.

Course Contents:

Introduction: MySQL, Oracle, SQL, Data types, SQL / PLSQL, Oracle Software Installation, User Type, Creating User , Granting. **Basic Parts of Speech in SQL:** Creating Newspaper Table, Select Command (Where, order by), Creating View, Getting Text Information & Changing it, Concatenation, Cut & paste string (RPAD, LPAD, TRIM, LTRIM, RTRIM, LOWER, UPPER, INIT, LENGTH, SUBSTR, INSTR, SOUNDEX). **Playing The Numbers:** Addition, Subtraction, Multiplication, Division, NVL, ABS, Floor, MOD, Power, SQRT, EXR, LN, LOG, ROUND, AVG, MAX, MIN, COUNT, SUM, Distinct, SUBQUERY FOR MAX, MIN. **Grouping things together:** Group By, Having, Order By, Views Renaming Columns with Aliases. **When one query depends upon another:** Union, Intersect, Minus, Not in, Not Exists. **Changing Data:** INSERT, UPDATE, MERGE, DELETE, ROLLBACK, AUTOCOMMIT, COMMIT, SAVEPOINTS, MULTI TABLE INSERT, DELETE, UPDATE, MERGE. **Creating And Altering tables & views:** Altering table, Dropping table, Creating view, Creating a table from a table. **By What Authority:** Creating User, Granting User, Password Management. **An Introduction to PL/SQL**: Implement a few problems using PL/SQL (e.g. Prime Number, Factorial, Calculating Area of Circle, etc.). **An Introduction to Trigger and Procedure:** Implement few problems using Trigger and Procedures. **An Introduction to Indexing:** Implement indexing using a large database and observe the difference of Indexed and Non-Indexed databases.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Demonstrate the knowledge in projects with a commercial relational database system (Oracle) and design a team-based project.
CO2	Utilize the database design principles, SQL and PL SQL.
CO3	Demonstrate the relational database theory and be able to write relational algebra expressions for queries.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Database System Concept, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fourth edition
2. Files and Databases- An Introduction, Peter D. Smith and G.M. Barnes, Addison- Wesley
3. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, Third edition

Course Title: Operating System and System Programming		**Credits:** 3.0
Course Code: 0613 3135	**Contact hours:** 3 hours/week	

Rationale:
This course presents fundamental concepts related to the design and implementation of operating systems. Topics include basic operating system structure, process scheduling, process and thread synchronization and concurrency, memory management, file systems.

Objectives:

- To acquaint students with the role of the operating system as a high level interface to the hardware.
- To make them understand basic idea about the low level implementation of CPU dispatch.
- To make them understand basic idea about the low level implementation of memory management.
- To provide knowledge about the performance trade-offs inherent in OS implementation

Course Contents:
Introduction: Operating Systems Concept, Computer System Structures, Operating System Structures, Operating System operations, Protection and Security, Special-Purpose Systems. **Fundamentals of OS:** OS services and components, multitasking, multiprogramming, time sharing, buffering, spooling **Process Management:** Process Concept, Process Scheduling, Process State, Process Management, Interprocess Communication, interaction between processes and OS, Communication in Client-Server Systems, Threading, Multithreading, Process Synchronization.**Concurrency control:** Concurrency and race conditions,

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3									2		
CO 2		3										
CO 3					3							

mutual exclusion requirements, semaphores, monitors, classical IPC problem and solutions, Dead locks - characterization, detection, recovery, avoidance and prevention. **Memory Management**: Memory partitioning, Swapping, Paging, Segmentation, Virtual memory - Concepts, Overlays, Demand Paging, Performance of demand paging, Page replacement algorithm, Allocation algorithms. **Storage Management**: Principles of I/O hardware, Principles of I/O software, Secondary storage structure, Disk structure, Disk scheduling, Disk Management, Swap-space Management, Disk reliability, Stable storage implementation. **File Concept:** File support, Access methods, Allocation methods, Directory systems, File Protection, Free Space management.**Protection & Security**: Goals of protection, Domain of protection, Access matrix, Implementation of access matrix, Revocation of access rights, The security problem, Authentication, One-time passwords, Program threats, System threats, Threat monitoring, Encryption, Computer-security classification. **Distributed Systems:** Types of Distributed Operating System, Communication Protocols, Distributed File Systems,Naming and Transparency, Remote File Access, Stateful Versus Stateless Service, File Replication. **Case Studies:** Study of a representative Operating Systems. **System Programming:** Introduction to System Programming and Linux / Unix, Shell Programming, C Language forSystem Programming, Make and Make files, Process and Signals, Threads, Inter process Communications, X- Window Programming, Principle of single and multi-user operating systems.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Compare the functionality of different computing hardware structures and Operating System Structures.
CO 2	Discuss issues of Process Management including process structure, synchronization, scheduling and communication.
CO 3	Discuss memory management issues including pages, segmentation and virtual memory, distributed system
CO 4	Explain the operation of various File Management Algorithms.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4					3			2				

Textbook

1. Operating System Concepts by Silberschatz& Galvin Wiley 2000 (7th Edition)
2. Operating Systems by Achyut S. Godbole Tata Mc Graw Hill (2nd Edition)

Course Title: Operating System and System Programming Lab		**Credits:** 1.5
Course Code: 0613 3136	**Contact hours:** 3 hours/week	

Rationale:
The goal of this course is to have students understand and appreciate the principles in the design and implementation of some of the features on operating systems software.

Objectives:

- To facilitate idea about Inter-process communication
- To facilitate the knowledge of CPU scheduling for processes
- To facilitate the knowledge of Process Synchronization using Critical section
- To facilitate the knowledge of Multi -Threading and Thread Synchronization
- To acquaint students with the knowledge of network operating system tasks through simulation/implementation

Course Contents:
Thread programming: Creating thread and thread synchronization. **Process Programming:** The Process ID, Running a New Process, Terminating a Process, Waiting for Terminated Child Processes, Users and Groups, Sessions and Process Groups. **Concurrent Programming:** Using fork, exec for multi-task programs. **File Operations:** File sharing across processes, System lock table, Permission and file locking, Mapping Files into Memory, Synchronized, Synchronous, and Asynchronous Operations, I/O Schedulers and I/O Performance. **Communicating across processes:** Using different signals, Pipes, Message queue, Semaphore, Semaphore arithmetic and shared memory.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Understand some features on Linux
CO 2	Understand Process to process communication, process synchronization, multi-threading
CO 3	Explain concept on managing multiprocessing system.
CO 4	Study on Network Operating System

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4										3		

Textbook

1. The 'C' Odyssey UNIX-The Open, Boundless C by Meeta Gandhi, Tilak Shetty, Rajiv Shah.
2. Beginning Linux Programming by Neil Matthew and Richard Stones
3. Linux System Programming by Robert Love

Course Title: Software Engineering and Design Patterns		**Credits:** 3.0
Course Code: 0613 3141	**Contact hours:** 3 hours/week	

Rationale:

This course is designed to provide a general introduction to software engineering and design. It will introduce the important concepts such as software processes and agile methods, essential software development activities from initial software specification through to system evolution. Students will learn on basic design principles and how those principles can be utilized to make more modular and scalable programs.

Objectives:

- To give students an insight about common software engineering processes and well-known practices.
- To make students understand the impact of requirement engineering and the proper way to do that.
- To make students understand basic design principles and how those principles can be utilized to make more modular and scalable programs.
- To help students develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain.
- To provide knowledge of basic software measurement concepts and how to allocate resources from the perspective of a software manager or team lead.
- To facilitate students with the knowledge how to properly test their software and modern software verification and validation practices.

Course Contents:

Introduction: Introduction to Software Engineering, Software Development Process and Various Life Cycle Models. **Requirement Analysis:** Communication Techniques, Analysis Principles, Software Prototyping, Requirement Specification. **Analysis Modeling:** Steps of system analysis, Feasibility study, Economic and technical analysis, System specification, the elements of analysis model, Data modeling, Functional modeling and information flow, Behavioral modeling, Mechanics of structured analysis, Data Dictionary. **Software Design:** Design principles, Design Concepts, effective modular design, design heuristics, Data Design, Architectural Design process, Transformation mapping, Transaction mapping, interface design, human-computer interface design, procedural design. **Software Testing:** Testing fundamentals, test case design, white-box testing, black-box testing, testing GUIs, Unit testing, Integration testing, validation testing, system testing, debugging. **Maintenance:** Major maintenance activities, estimating maintenance cost and productivity. **Technical Metrics for Software:** Software quality, Framework for technical metrics, metrics for analysis and design models, source code, testing and maintenance. **Software Architecture:** Pipe and Filter, Object Oriented, Event Based, Layered System, Data-centered repository, Process Control Architectures. **Software Project Management:** Cost estimation, risk analysis, project scheduling. **Design Patterns:** Introduction to design patterns. Different Patterns: Strategy, Observer, Factory, Singleton, Command, Adapter, Facade, Template Method, Iterator, Composite, State, Proxy, Compound Patterns. **Formal Methods:** Formal Methods in Software Engineering: its need and application, Formal specifications, Formal Verifications, Introduction to Z Language, Formal methods and testing.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Execute common software engineering processes and well-known practices.
CO2	Apply basic design principles and design pattern to write modular codes
CO3	Understand modern software verification and validation practices
CO4	Exercise project management process while developing software.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Software Engineering: A Practitioner's Approach- Roger S. Pressman.
2. Head First Design Patterns, Eric & Elisabeth Freeman, O'REILLY.

Course Title: Software Engineering and Design Patterns Lab	**Cr.:** 1.5
Course Code: 0613 3142	**Contact hours:** 3 hours/week

Rationale:

The course attempts to foster an understanding of software development process and practices. This can be done through the use of a team project running throughout the course. They will work on all stages of software development life cycles and apply the knowledge with a development of prototype. In this course the team will produce all type of documentations following industrial template.

Objectives:

- To provide students a hands-on training on basic design principles and how those principles can be utilized to make more modular and scalable programs.
- To familiarize students with basic software engineering diagrams like (class diagram, state diagram, use-case diagrams, etc.) and how these diagrams can be used to describe a software from different viewpoints.
- To help students develop the ability of significant teamwork and project based experience
- To help develop skills that will enable the students to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain

Course Contents:

Software Engineering lab work is solely designed to attain hands-on experience of architectural design, documentation and testing of software so that students can develop the software following the documents only. Also this lab includes Introduction to UML, Introduction to CASE Tools and Introduction to MVC Pattern. **Step1 (Requirement Engineering)**: Choose a company/institute/client for which software will be developed (make sure that they will provide required information whenever necessary). Follow the steps for eliciting requirements and generate a use-case diagram. Also analyze the sufficiency of the requirement engineering outcome for steps to follow. **Step 2 (Analysis model to Architectural and Component level design)**: Generate Activity diagram, Data flow diagram(DFD), Class diagram, State diagram, Sequence diagram and follow other relevant steps for creating complete architectural and component level design of the target software. **Step 3 (User Interface design, Design evaluation, Testing strategies and Testing Tactics)**: Perform the user interface design with the help of swim lane diagram. Carry out the design evaluation steps. Generate all test cases for complete checking of the software using black box, white box testing concept. **Step 4 Software testing and debugging. Step 5 (Managing Software Projects)**: Analyze the estimation and project schedule.

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3									
CO 3	3											
CO 4											3	

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Practice software engineering development life cycle stages in application development
CO2	Produce Software Requirement Specification, Design Specification and Test Cases according to industry standard template
CO3	Conduct design, evaluation and testing.
CO4	Exercise project management process while developing software.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Software Engineering: A Practitioner's Approach- Roger S. Pressman.
2. Head First Design Patterns, Eric & Elisabeth Freeman, O'REILLY.

Course Title: Artificial Intelligence		**Credits:** 3.0
Course Code: 0619 3193	**Contact hours:** 3 hours/week	

Rationale:

Web search, speech recognition, face recognition, machine translation, autonomous driving, and automatic scheduling; these are all complex real-world problems, and the goal of artificial intelligence (AI) is to tackle these with rigorous tools. This course will help students to learn the foundational principles that drive these applications and practice implementing these systems. The main goal of the course is to equip students with the tools to tackle new AI problems they might encounter in life. This course will make students able to build applied systems and to account for intelligence from a computational point of view by introducing representations, techniques, and architectures used.

Objectives:

- To provide the most fundamental knowledge to the students so that they can understand what the AI is
- To facilitate with knowledge of theoretic proofs and formal notations of AI
- To introduces students to the basic knowledge representation, problem solving, and learning methods of artificial intelligence
- To help them develop intelligent systems by assembling solutions to concrete computational problems
- To make them understand the role of knowledge representation, problem solving, and learning in intelligent-system engineering, and appreciate the role of problem solving, vision, and language in understanding human intelligence from a computational perspective
- To help them explore applications of rule chaining, heuristic search, logic, constraint propagation, constrained search, and other problem-solving paradigms

Course Contents:

What is Artificial Intelligence: The AI problems, The underlying assumption, What is an AI technique. **Problems, Problem spaces and**

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3				2				1			
CO 2			3						2	1		
CO 3					3				2			
CO 4									2		3	

Search: Defining the problem as a state space search, Production system, Problem characteristics. **Heuristics Search Techniques:** Generate and Test, Hill climbing, Best First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis. **Knowledge Representation Issues:** Representation and Mappings, Approaches to knowledge Representation, Issues in Knowledge representation. **Using Predicate logic:** Representing simple facts in logic, Representing Instance and Isa relationships, Computable functions and Predicates, Resolution. **Representing Knowledge using Rules:** Procedural versus Declarative Knowledge, Logic Programming, Forward versus Backward Reasoning, Matching. **Game playing:** Overview, The Minimax Search Procedure, Adding Alpha-Beta cutoffs, Additional refinements, iterative Deepening. **Planning:** Overview, An example Domain: The Blocks World, Components of a planning system, Goal stack planning. **Understanding:** What is Understanding, What makes Understanding hard, Understanding as constraint satisfaction. N**atural Language Processing:** Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing. **Expert systems:** representing and using domain knowledge, Expert system shells explanation, Knowledge Acquisition.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Compare AI with human intelligence and traditional information processing, and analyze AI-related problems to choose solution techniques precisely.
CO 2	Examine different methods for knowledge, facts and logic representation, manipulation and transformations.
CO 3	Examine AI knowledge in a specific structured domain of game playing.
CO 4	Construct plans to solve nontrivial problems by combining different search and decomposition techniques, and understand the generation and translation of knowledge in the domain of NLP.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P10	P11	P12
CO 1	3	3										
CO 2			3									
CO 3				3								
CO 4						2				3		1

Textbook
1. Artificial Intelligence: A Modern Approach by Stuart Russel.
2. The Cambridge Handbook of Artificial Intelligence by Keith Frankish, William M. Ramsey.

Course Title: Artificial Intelligence Lab	**Credits:** 1.5
Course Code: 0619 3194	**Contact hours:** 3 hours/week

Rationale:
This course is offered to help students to learn the foundational principles that drive these applications and practice implementing these systems. Course is designed with the intention to equip students with the tools to tackle new AI problems they might encounter in life. Students completing this course are supposed to build intelligent agent systems from a computational point of view.

Objectives:
- To make them understand functionality of intelligent agents
- To provide basic idea about how intelligent agents solve different real-world problems
- To make them solve different AI problem

Course Contents:
Students will have to understand the functionalities of intelligent agents and how the agents will solve general problems. Students have to use a high-level language (Python, Prolog, LISP) to solve the following problems: **Backtracking:** State space, Constraint satisfaction, Branch and bound. Example: 8-queen, 8- puzzle, Crypt-arithmetic. **BFS and production:** Water jugs problem, The missionaries and cannibal problem. **Heuristic and recursion:** Tic-tac-toe,Simple bock world, Goal stack planning, The tower of Hanoi. **Question answering:** The monkey and bananas problem.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Develop artificially intelligent systems that can deal with knowledge and logic presentation and manipulation
CO 2	Develop artificially intelligent programs to solve brain storming puzzle games
CO 3	Develop simple games that are artificially intelligent
CO 4	Design, develop, examine and evaluate an intelligent agent which solves a domain specific precise problem which usually does a human expert

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								

Text Books:
1. Artificial Intelligence; Elaine Rich and Kevin Knight
2. Artificial Intelligence; Winston, Patrick Henry.

Course Title: Microprocessor and Embedded System	**Credits:** 3.0
Course Code: 0714 3177	**Contact hours:** 3 hours/week

Rationale:
This course will provide the student with a fundamental understanding of microprocessor architecture and also relate how the microcontroller differs with microprocessor. Learn about the RISC and CISC, ARM Cortex, SAP microprocessor. Able to design application oriented embedded systems. And finally get insight about various communication protocols to interface different peripheral devices for both microprocessor and microcontroller

Objectives:

- Provide students with a basic background about Microprocessors.
- Can differentiate Microprocessor and Microcontroller.
- Understand different communication protocols used for embedded devices.

Course Contents:

Introduction to microprocessors and embedded systems: History, evolution, classification and characteristics of microprocessors and embedded systems, Difference between microprocessor and microcontroller, Examples of embedded systems in various domains such as IoT, robotics, automotive, etc.
Simple As Possible (SAP) microprocessors: Architecture, instruction set, programming, fetch cycle, execution cycle, microprogramming, register structure of SAP-1, SAP-2, and SAP-3 microprocessor.
8086/8088/Arm Cortex M/RISC-V microprocessors: Internal architecture, register structure, instruction set and format, programming in machine or assembly languages, interrupt structure, memory organization and segmentation, DMA, I/O operation, addressing, uses of flags, microprocessor interface ICs, peripheral interfacing, microprocessor based system design, coprocessor, multiprocessor system.
Embedded System Basics: Definition and characteristics of embedded systems, Embedded system design challenges and constraints, Introduction to real-time operating systems (RTOS), Hardware and software considerations in embedded systems.
Microcontroller Peripherals and Interfacing Protocols: Overview of microcontroller peripherals, Configuring and programming timers for various applications (PWM, interrupts, etc), Communication protocols (UART, USART, RS-232, SPI, I2C, RJ-45 etc.), Interfacing with external devices (LEDs, Switches, 7 segment displays, LCDs, DC motors, Stepper motors, Servo motors), Analog-to-digital conversion and sensor interfacing.
Embedded System Design: Strategies for effective embedded system design, Choosing the right microcontroller for a specific application, Low-power design techniques, Case studies of real-world embedded system projects

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	Understand the difference between general purpose microprocessor and microcontroller.
CO 2	Interpret microprocessor‘s and microcontroller‘s internal architecture and their operation.
CO 3	Interpret how low-level programming language works in a machine such as assembly language and C language.
CO 4	Understand and evaluate various types of interfacing protocols.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3		3	2									
CO 4			3									

Mapping of Course Learning Outcomes (COs) with the Teaching-Learning & Assessment Strategy

CO	Teaching Learning Strategy	Assessment Strategy
CO 1	Lectures, Reading Material	Assignment, Final Exam
CO 2	Lectures	Class Test, Final Exam
CO 3	Lectures	Class Test, Final Exam
CO 4	Lectures, Reading Material	Class Test, Final Exam

Textbook

1. Digital Computer Electronics by Albert P. Malvino and Gerald A. Brown
2. Microprocessors and interfacing by Douglas V hall
3. Microprocessors, PC Hardware and interfacing by N. Mathivanan
4. Computer Organization and Design – The Hardware / Software Interface by David A. Patterson and John L. Hennessy
5. Digital Design and Computer Architecture by Sarah Harris, David Harris
6. Assembly Language Programming and Organization for the IBM PC by Ytha Yu and Charles Marut
7. The Intel Microprocessors by Barry B Brey
8. Microprocessors and Microcomputer- based system design by Mohamed Rafiquzzaman

Course Title: Microprocessor and Embedded System Lab		**Credits:** 1.5
Course Code: 0714 3178	**Contact hours:** 3 hours/week	

Rationale:

This course will provide the student with a fundamental understanding between the interplay of machine code and microprocessor. Able to design application oriented embedded systems. And finally get insight about various communication protocols to interface different peripheral devices for both microprocessor and microcontroller.

Objectives:

- Provide students with a basic background about Microprocessors.
- Can differentiate Microprocessor and Microcontroller.
- Understand different communication protocols used for embedded devices and use them while designing an embedded system.

Course Contents:

Lab 1: Introduction to Microcontroller Development Tools
Lab 2: Motor (DC motor, Stepper motor, Servo motor, etc.) Interfacing
Lab 3: Study on Hardware and Software Interrupts
Lab 4: Timers and PWM
Lab 5: Serial Communication (Hardware and Software Serial Communication)
Lab 6: Analog-to-Digital and Digital-to-Analog Conversion (Sensor Interfacing)
Lab 7: Study on I2C, SPI, and RJ-45 Communication protocols
Lab 8: Interfacing Peripheral devices using MATLAB
Lab 9-12: Project Work
Lab 13: Project Presentation
Lab 14: Lab final exam

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Understand software at machine level.
CO 2	Understand and analyze functions of different peripheral devices interfaced with microprocessor.
CO 3	Analyze and understand the working of different practical embedded systems.
CO 4	Design microprocessor/microcontroller based system as part group project.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3		3										
CO 4			3									

Mapping of Course Learning Outcomes (COs) with the Teaching-Learning & Assessment Strategy

CO	Teaching Learning Strategy	Assessment Strategy
CO 1	Lab class, Experiments	Continuous assessment, Final Exam
CO 2	Lab class, Experiments	Continuous assessment, Final Exam
CO 3	Lab class, Experiments	Continuous assessment, Final Exam
CO 4	Lab class, Experiments	Continuous assessment, Final Exam

Textbook

- Raspberry Pi Cookbook by Simon Monk
- Embedded Systems with ARM Cortex-M Microcontrollers with Assembly Language and C by Yifeng Zhu
- Microprocessors, PC Hardware and interfacing by N. Matł

Course Title: Capstone Project Work III	**Credits:** 3.0
Course Code: 0610 3250	**Contact hours:** 6 hours/week

Rationale:

This is a very important course that intends to make students prepared to face challenges in relevant industry.

Objectives:

- To facilitate necessary knowledge about latest technology
- To help them develop skills on software development
- To help them develop skills on teamwork and presentation

Course Contents:
Projects must possess innovative ideas which reflect contemporary IT trends. Supervisor has to ensure that every accepted project contains a basic level of research work. Projects that meet the software/hardware requirements of SUST or any other IT organization are highly preferable. Students have to give a presentation on their project works. Departments should take appropriate steps to archive all the projects and keep tracks to maintain the genuineness of the projects.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2					3							
CO 3			3							3		
CO 4										3		

CO 1	Apply latest state of the art technologies
CO 2	Design and implement ideas for complete software
CO 3	Evaluate existing computer and mobile applications
CO 4	Explain ideas to groups and present their noble findings

Mapping of Course Learning Outcomes to Program Learning Outcomes

Course Title: Technical Writing And Presentation	**Credits: 1.5**
Course Code: 0611 3230	**Contact hours:** 3 hours/week

Rationale:
In this course students will be facilitated with knowledge on interpretation of their technical knowledge through writing. They will learn how to write in a specific format using the latest technologies, draw their diagrams and also present their work in front of the audience.

Objective:
- To facilitate necessary knowledge about methods for technical writing
- To acquaint students with basic tools for writing, presentations and drawings
- To help them enhance the skills on presentation and communication

Course Content:
Issues of technical writing and effective oral presentation in Computer Science and Engineering; Writing styles of definitions, propositions, theorems and proofs; Research Methodologies; Preparation of reports, research papers, theses and books: abstract, preface, contents, bibliography and index; Writing of book reviews and referee reports; Writing tools: LATEX; Diagram drawing software; presentation tools.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Apply skills on technical writing for writing technical and academic reports
CO 2	Use latest technologies for writing and drawing
CO 3	Apply skills on designing graphical representations
CO 4	Develop academic and industrial level of presentation skills

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook
1. Engineers' guide to technical writing - Kenneth G. Budinski
2. Writing for Computer Science - Justin Zobel

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2					2							
CO 3			3							3		
CO 4										2		

Course Title: Machine Learning and Neural Network	**Credits:** 3.0
Course Code: 0619 3293	**Contact hours:** 3 hours/week

Rationale:

Current digital world is totally driven by data and information. Prediction, recommendation, identification and many other activities are dependent on data mining, summarization and big data analysis. And for completing these types of works the importance of Machine Learning is increasing day by day. This course is intended to introduce students to the world where machines can learn and act based on learning like how human beings work.

Objectives:

- To familiarize with a set of well-known supervised, unsupervised and semi-supervised learning algorithms
- To help them Understand how machine learning algorithms are evaluated
- To make them able to formulate machine learning problems corresponding to different applications
- To introduce with the basic theory underlying machine learning
- To help them to apply machine learning algorithms to solve problems of moderate complexity.

Course Contents:

Introduction to Machine Learning Concepts: Concepts of ML. Types of Machine Learning, Some ML applications and examples. The main components of a ML system. Requirements to design a ML system. Testing ML algorithms, Linear Regression, Logistic Regression, Regularization, Decision Tree, Learning a concept and hypothesis, Naïve Bayes Classifier, Artificial Neural Network, Linear Discriminants, Perceptron Learning, Delta Rule, Multi-layer Neural Network, Back-propagation Algorithm, Unsupervised Learning, Clustering Technique, K-means Clustering, Clique Graph, Hierarchical Clustering, Anomaly Detection, Dimensionality Reduction, N-gram Model, Hidden Markov Model, Support Vector Machine, Genetic Algorithm, Reinforcement Learning, Information Retrieval, Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing. Recommender System. Deep Learning.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Discuss about fundamental topics of ML, its classification, presentation, components and its impact on the current and future world.
CO 2	Examine different types of machine learning approaches for solving different types of real life problems.
CO 3	Understand methods like principle component analysis, dimensionality reduction, feature extraction to prepare and manipulate data to make them suitable for ML approaches.
CO 4	Identify a real life problem and design and develop computer based systems implementing machine learning models based on real life data from the identified domain.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3				3								
CO 4				3								2

Textbook

1. Machine Learning, An Algorithmic Perspective(2nd Edition), Stephen Marsland
2. Introduction to machine learning (2nd edition), Alpaydin, Ethem.
3. The Art and Science of Algorithms that Make Sense of Data Machine Learning, Peter Flach.
4. Machine Learning, Tom Mitchell, McGraw Hill.

Course Title: Machine Learning and Neural Network Lab		**Credits:** 1.5
Course Code: 0619 3294	**Contact hours:** 3 hours/week	

Rationale:

Current digital world is totally driven by data and information. Prediction, recommendation, identification and many other activities are dependent on data mining, summarization and big data analysis. And for completing these types of works the importance of Machine Learning is increasing day by day. This course is intended to introduce students to the world where machines can learn and act based on learning like how human beings work.

Objectives:

- To familiarize with a set of well-known supervised, unsupervised and semi-supervised learning algorithms
- To help them Understand how machine learning algorithms are evaluated
- To make them able to formulate machine learning problems corresponding to different applications
- To help them to apply machine learning algorithms to solve problems of moderate complexity

Course Contents:

Laboratory works based on CSE475. ID3 Algorithm for Decision Tree, Regression using LSE and estimating MSE, kNN Algorithm as Nearest Neighbor Classifier, Apply NB Classifier for a Classification Task. Application of the MLP-BP ANN algorithm, Application of GA for solving a problem, Application of SVM, Application of HMM, Exclusive clustering: K- means algorithm, Agglomerative clustering: Hierarchical algorithm.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Discuss about fundamental topics of ML, its classification, presentation, components and its impact on the current and future world.
CO 2	Examine different types of machine learning approaches for solving different types of real life problems.
CO 3	Understand methods like principle component analysis, dimensionality reduction, feature extraction to prepare and manipulate data to make them suitable for ML approaches.
CO 4	Identify a real life problem and design and develop computer based systems implementing machine learning models based on real life data from the identified domain.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3				3								
CO 4				3								2

Textbook

1. Machine Learning, An Algorithmic Perspective(2nd Edition), Stephen Marsland
2. Introduction to machine learning (2nd edition), Alpaydin, Ethem.
3. The Art and Science of Algorithms that Make Sense of Data Machine Learning, Peter Flach.
4. Machine Learning, Tom Mitchell, McGraw Hill.

Course Title: Data Communication	**Credits:** 2.0
Course Code: 0714 3273	**Contact hours:** 2 hours/week

Rationale:

The main course is to infer the working knowledge of data transmission concepts, line control and line sharing and also is to understand the operation of compression optimizing data transfer algorithms.

Objective:

1. To familiarize with modern telecommunications and the architecture of a number of different networks.
2. To impart knowledge on protocol layering and different multiplexing techniques, data compression algorithms to optimize network bandwidth.
3. To familiarize with the use reliability, redundancy and availability of different techniques to meet network performance criteria.

Course Content:

Introduction: Communication Models, Communication Network Standards and Organization, Introduction to TCP/IP Models. Data Transmission Basics: Analog and Digital Data, Spectrum and Bandwidth, Transmission Impairments, Data Rate, and Channel Capacity. Data Encoding: NRZI, Manchester and Differential Manchester Encoding, ASK, FSK, PSK, QPSK, QAM Encoding, Pulse Code Modulation, Delta Modulation. Data Transmission: Asynchronous and Synchronous Data Transmission Techniques. Analog Transmission: Digital-To-Analog Conversion, Amplitude/Frequency/Phase Shift Keying, Quadrature Amplitude Modulation, Analog-to-Analog Conversion, Amplitude/Frequency/Phase

Modulation. Multiplexing: Frequency-Division Multiplexing, Wavelength-Division Multiplexing, Synchronous Time-Division Multiplexing, Statistical Time-Division Multiplexing, Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum. Transmission Media: Twisted-Pair Cable, Coaxial Cable, Fiber-Optic Cable, Radio Waves, Microwaves, Infrared. Error Detection and Correction: Redundancy, Parity Cheeks, Hamming Distance, CRC Error Correction, Checksum. Multiple Access: ALOHA, CSMA, CSMALCD, CSMALCA, FDMA, TDMA, CDMA. Wired LANs: Ethernet, IEEE Standards, Standard Ethernet, IEEE 802.11, Bluetooth. Connecting Devices: Passive Hubs, Repeaters, Active Hubs, Bridges, Two-Layer Switches, Routers, Three-Layer Switches, Gateway, Backbone Networks, Virtual LANs.

Course Learning Outcome:

After the successful completion of the course, the student will be able to-

CO 1	Explain fundamental concepts of data communication and its components, different types of network, protocol layering and responsibilities of different layers for different protocols
CO 2	Percept the digital and analogue representations of signals and analyze the mechanism of encoding schemas
CO 3	Analyze resource sharing techniques called multiplexing for transmitting multiple signals through a single channel and identify techniques of different types of multiplexing.
CO 4	Identify and analyze principles of security, performance and reliability of different networks.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											3
CO 2	3											
CO 3		3										
CO 4		3		3								

Textbook

1. Data Communications and Networking by Behrouz A. Forouzan

Course Title: Digital Signal Processing	**Credits:** 3.0
Course Code: 0714 3275	**Contact hours:** 3 hours/week

Rationale:

This course is designed to equip students with an understanding of the principles and techniques needed to analyze, design and implement digital signal processing systems and algorithms, which are vital in fields such as telecommunications, audio and image processing, control systems and many more.

Objectives:

- To provide students with fundamental knowledge and techniques in signal processing
- To teach students how to analyze and design digital signal processing systems.
- To equip students with the ability to apply digital signal processing techniques to real-world problems, such as speech and image processing
- To Design, implementation, analysis and comparison of digital filters for processing of discrete time signals
- To prepare students to pursue advanced studies or professional careers in digital signal processing and related fields.

Course Contents:

SIGNALS AND SYSTEMS: Basic elements of DSP – concepts of frequency in Analog and Digital Signals – sampling theorem –Discrete – time signals, systems – Analysis of discrete time LTI systems – Z transform – Convolution (linear and circular) – Correlation. **FREQUENCY TRANSFORMATIONS:** Introduction to DFT – Properties of DFT – Filtering methods based on DFT – FFT Algorithms Decimation – in – time Algorithms, Decimation – in – frequency Algorithms – Use of FFT in Linear Filtering – DCT. **IIR FILTER DESIGN:** Structures of IIR – Analog filter design – Discrete time IIR filter from analog filter – IIR filter design by Impulse invariance, Bilinear transformation, Approximation of derivatives – (HPF, BPF, BRF) filter design using frequency translation. **FIR FILTER DESIGN:** Structures of FIR – Linear phase FIR filter – Filter design using windowing techniques, Frequency sampling techniques – Finite word length effects in digital Filters. **APPLICATIONS:** Multirate signal processing – Speech compression – Adaptive filter – Musical sound processing – Image enhancement.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Explain the basic concepts of analog and discrete-time signals.
CO2	Classify systems based on linearity, causality, shift-variance, stability criteria and represent transfer function of the selected system.
CO3	Evaluate system response of a system using convolution methods, Z-transform, DFT, DCT, DIF-FFT or DIT-FFT algorithms, windowing techniques.
CO4	Design FIR and IIR digital filters.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4/e, Pearson Education, 2007.
2. A.V. Oppenheim, R. W. Schafer, Discrete-Time Signal Processing, 3/e, Prentice Hall of India, 2009.
3. Steven W. Smith,Digital Signal Processing: A Practical Guide for Engineers and Scientists, Newnes, 2002
4. S. Mitra, Digital Signal Processing-A computer based approach, McGraw-Hill Companies, 2006

Course Title: Digital Signal Processing Lab	**Credits:** 1.5
Course Code: 0714 3276	**Contact hours:** 3 hours/week

Rationale:

The course will provide students with hands-on experience through the use of software tools and programming languages such as Python or Matlabto implement and analyze digital signal processing algorithms.

Objectives:

- To provide students with hands-on experience on different principles and techniques in signal processing.
- To teach students how to analyze, design and implement digital signal processing systemsto real-world problems such as speech and image processing.
- To familiarize students with software tools, such as Python or MATLAB, that can be used to implement digital signal processing algorithms and analyze the results.
- To Design, implementation, analysis and comparison of digital filters for processing of discrete time signals
- To prepare students to pursue advanced studies or professional careers in digital signal processing and related fields.

Course Contents:

Introduction to programming language Python/ Matlab and software tools necessary for digital signal processing; Generating elementary signals (e.g., sinusoids, unit impulse, ramp, exponential signals etc.) and visualizing them; Building simple systems using basic operations (e.g., time shifting, addition, multiplication) on signals; Developing systems utilizing the computation of convolution, cross-correlation, auto-

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2	3											
CO 3	3											
CO 4			3									

correlation, etc.; Building systems for computation of z-transform, Discrete Fourier Series, Discrete Fourier Transform (DFT), different algorithms for Fast Fourier Transforms and Discrete Cosine Transform (DCT); Designing digital Filters and apply on natural signals like speech,Speech signal processing and visualization; Multirate digital signal processing (e.g. up-sampling, down-sampling, etc.).

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Experiment with sequence generation, correlation, and convolution of Discrete-Time signals.
CO2	Interpret discrete-time signals using various transforms like DFT and DCT.
CO3	Apply FFT algorithms for various signal processing operations.
CO4	Analyze the IIR and FIR digital filters.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3				3							
CO 2	3				3							
CO 3	3				3							
CO 4					3							

Textbook

1. J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4/e, Pearson Education, 2007.
2. A.V. Oppenheim, R. W. Schafer, Discrete-Time Signal Processing, 3/e, Prentice Hall of India, 2009.
3. Steven W. Smith, Digital Signal Processing: A Practical Guide for Engineers and Scientists, Newnes, 2002
4. S. Mitra , Digital Signal Processing-A computer based approach, McGraw-Hill Companies, 2006
5. Allen B. Downey,Think DSP- Digital Signal Processing in Python,O'Reilly Media, 2016

Course Title: Computer Architecture	**Credits:** 3.0
Course Code: 0714 3279	**Contact hours:** 3 hours/week

Rationale:

This course is designed to provide a strong foundation for students to understand the modern areas of computer architecture. It will describe a broad range of architectural designs highlighting the design decisions and how these design decisions impact on system performance. The students will be able to apply these insights and principles to future computer designs.

Objectives:

- To make the students recognize the fundamental technologies and performance evaluation of different computer systems.
- To help them to know the instruction set architecture of a system and variations of ISA in different systems.
- To describe how computer performs arithmetic operations.

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1		3										
CO 2			3	3					3			
CO 3			3						3			
CO 4					3				3			

- To provide ideas about internal architecture of a processor along with parallel computing.
- To identify the underlying technologies on different levels of memory hierarchy and their management in a system.
- To accumulate basic ideas about fundamental technologies on multicore and multiprocessing system and their application.

Course Contents:

Introduction to Computer Architecture: Overview and history; Cost factor; Performance metrics, Fundamental blocks of computer. **Instruction set architecture: C**lassifying instruction set architectures, Registers, Addressing Modes, RISC versus CISC, x86 Architecture, ARM Architecture. **Memory Hierarchy:**Hierarchical Organization, Cache memory; Basic cache structure and design; Fully associative, direct, and set associative mapping; Analyzing cache effectiveness; Replacement policies; Writing to a cache; Multiple caches; Upgrading a cache; Main Memory; Virtual memory and machine, Paging, Replacement strategies. **Data Representation:** Data type representation, signed number, fixed point, floating point, character. **Processor and Pipelining:**Datapath, pipelined Datapath,Pipelining basics, types, stalling and forwarding, Throughput and Speedup of Pipelining, Pipelining hazards. **Parallelism:** Instruction level parallelism, introduction, challenges and limitations, Scalar and superscalar pipelining, branch prediction, increase uniprocessor throughput.**Multiprocessors and Multi-core Computers:** SISD, SIMD, and MIMD architectures; Centralized and distributed shared memory-architectures; Multi-core Processor architecture. **Input/output Devices:** Performance measure, Types of I/O device, Buses and interface to CPU, RAID.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Identify the fundamental technologies incorporated in computer architectures.
CO 2	Elevate the memory management technologies.
CO 3	Implement pipelining mechanism and parallel computing in to the processor.
CO 4	Improve I/O performance.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Computer Architecture and Organization by John P.Hayes.
2. Computer Organization and Design: The hardware / software interface by David A.Patterson and John L.Hennessy.

Fourth Year

Course Title: Final Year Project I	**Credits:** 2.0
Course Code: 0610 4150	**Contact hours:** 4 hours/week

Rationale:

This course is intended to provide students hands-on experience in identifying a real-life computing problem on contemporary topics related to their field of interest and find solution through design, development and validation.

Objectives:

- To facilitate students apply theoretical concepts in practical settings.
- To help students develop problem-solving skills.
- To help students develop communication and teamwork skills.
- To help students develop project management skills.
- To help students promote an understanding of professional and ethical responsibilities.

Course Contents:

Project work based on the core courses studied in the previous semesters.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Identify a real-life problem that can be translated to an engineering and/or computing solution through design, development and validation.
CO2	Identify outcomes and functional requirements of the proposed solution considering software and/or hardware specification and standards
CO3	Identify sub-components of a complex problem, prepare timeline and appropriate budget using the project management skill
CO4	Analyze, design, build, and evaluate engineering/computing system/subsystem with given specifications and requirements
CO5	Identify and validate the impact of environmental considerations and the sustainability of a system/subsystem of a complete project
CO6	Assess professional, ethical, and social impacts and responsibilities of the design project.
CO7	Function effectively in a multi-disciplinary team
CO8	Use modern analysis and design tools in the process of designing and validating of a system and subsystem
CO9	Present design project results through written technical documents and oral presentations

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2	3	2										
CO 3											3	
CO 4			3	3								
CO 5							3					
CO 6						3		3				
CO 7									3			
CO 8					3							
CO 9										3		

Course Title: Thesis I	Credits: 2.0
Course Code: 0610 4180	**Contact hours:** 4 hours/week

Rationale:
This course is intended to develop critical thinking and analytical skills in identifying, designing, conducting, and reporting research works on contemporary topics.

Objectives:
- To encourage students foster creativity and innovation
- To help students conduct literature review on current research
- To help students learn basic idea about research methodologies
- To help develop knowledge about research proposal
- To help develop skill of presentation
- To promote an understanding of ethical and professional considerations in research.

Course Contents:
Research work based on the core courses studied in the previous semesters.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Identify and formulate a research question through literature review on state-of-the-arts.
CO2	Competence in using appropriate research methods and techniques.
CO3	Prepare timeline and appropriate budget using the research project management skill.
CO4	Collect, analyze, and interpret data.
CO 5	Identify and validate the impact of environmental considerations and the sustainability of a system/subsystem of a research project
CO 6	Assess professional, ethical, and social impacts and responsibilities of the research project.
CO 7	Ability to collaborate with others and work effectively in a team.
CO 8	Use modern analysis and design tools in the process of developing and validating of a system
CO 9	Present research results through written technical documents and oral presentations.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2	3											
CO 3											3	
CO 4			3	3								
CO 5							3					
CO 6						3		3				
CO 7									3			
CO 8					3							
CO 9										3		

Course Title: Industrial Training/Attachment/Tour	Credits: 3
Course Code: CSE 0613 4120	**Contact hours:** 6 hours/week
Course Type: Field Work	

Rationale:
This course is specially designed to help the students to gain some extensive industry experience before graduating from the university. Software Engineering students need to be extremely skilled at using the latest technologies and to be able to work within a team. This course will help the students to understand how the job is actually done in an industry. They will be able to have the experience to work within a team to achieve a common goal. Also this course will help them to contribute to a real life project as they will work closely with a company for six months.

Objectives:
Objectives:
- To help the students get exposed to real work within an actual company rather than a mere demo project.
- Allowing the students to understand the work ethics of a company.
- Facilitate the students in building up their teamwork so that they can work efficiently within a team to achieve a common goal.

- Allowing the students to understand the skills that are demanded or required by the industry.

To let the students have work experience even before graduation.

Course Contents:

Semester long real world software development experience. Reporting and presentation after the internship.

Course Learning Outcomes (COs):

After successful completion of the course, students will be able to:

CO 1	apply software engineering and hardware related principles and best practices in a real-world work environment.
CO 2	communicate effectively with team members and stakeholders in a professional setting.
CO 3	analyze and solve complex problems using critical thinking and technical skills.
CO 4	demonstrate professionalism, ethical behavior, and responsibility in the workplace.

Mapping of COs with Program Learning Outcomes (POs):

Mapping of Course Learning Outcomes to Program Learning Outcomes

CO/PO	PO1	PL02	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2		1		2			2	3
CO2	3	2	2	2	2				3	1		3
CO3	3					2	1			2		3
CO4	3		1					1	1		2	3

Mapping Course Learning Outcomes (COs) with the Teaching-Learning and Assessment Strategy:

COs	Teaching-Learning Strategy	Assessment Strategy
CO1	CL, T, OR, GD	A, LE
CO2	CL, T, OR, GD, PrbL, PjrL	A, LE, RW
CO3	CL, T, OR, PrbL, PjrL	A, PP, Prj
CO4	GD, PrbL, PrjL, BL	V, P, RW, Prj

(CL = Class Lectures, T = Textbook, OR = Online Resources, GD = Group Discussion, PrbL = Problem-based Learning, PrjL = Project-based Learning, BL = Blended Learning)

(A = Assignment, V = Viva-voce, P = Presentation, RW = Report Writing, LE = Lab Examination, PP = Programming Problems, Prj = Projects)

Course Title: Computer and Information Security	**Credits:** 3.0
Course Code: 0612 4161	**Contact hours:** 3 hours/week

Rationale:

This is an introductory course on computer security. The main objective of this course is to introduce the basic concepts of cryptography and computer security covering physical security, operating system security as well as network and web security.

Objectives:

- To facilitate the basic knowledge of classic crypto systems and basic crypto primitives
- To assist students in developing introductory knowledge about block cipher and their different modes
- To help students conceptualize basic theories of different cryptographic mechanism such as symmetric and public key encryption, digital signature and hash function
- To assist students in developing basic knowledge about different security aspects covering multiple domains such as physical security, OS security, network security and web security
- To provide the basic knowledge of blockchain systems

Course Contents:

Basic terminology and security concepts: Fundamental concepts, Access control models, Cryptographic concepts, Security principles.**Classic Crypto Systems:** Substitution cipher, Vigenère cipher, Hill Cipher, One-time pads.**Symmetric Encryption:** Advanced Encryption Standard (AES) **Public Key Encryption:** RSA and ElGamal crypto systems.**Other crypto mechanisms**: Hash Function, Digital Signature.**Physical security**: Authentication technologies, Direct attacks, Physical Intrusion Detection.**Operating Systems Security:** Process, security, Memory and file system security, Application program security.**Malware and forensic analysis:** Insider & Malware attacks, Computer viruses, Privacy-invasive software, Countermeasures, Malware forensic.**Network Security:** Network security concepts, Vulnerabilities in Link, Network, Transport and Application layers, Firewall, Tunnelling and Intrusion detection, Denial of Service attacks, Countermeasures **Web security:** Attacks on clients, Attacks on servers, Countermeasures.**Blockchain and Bitcoin:** History of money, The need of decentralization, State machine replication, Concepts of transaction, block, blockchain and distributed consensus of Blockchain security, Blockchain applications.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Explain the Basic concepts of security and its different properties
CO 2	Explain the theoretical foundation of cryptography
CO 3	Demonstrate the practical application of cryptography in different application domains
CO 4	Analyze and identify security vulnerabilities in practical systems

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											3
CO 2	3											3
CO 3		3										
CO 4				3								

Textbooks

1. Introduction to Computer Security by Michael T. Goodrich and Roberto Tamassia
2. Introduction to Computer Security by Matt Bishop

Course Title: Computer and Information Security Lab	**Credits:** 1.5
Course Code: 0612 4162	**Contact hours:** 3 hours/week

Rationale

In this course, these students will carry out a number of hands-on lab works based on concepts gained in its counterpart theory course, 0612 4161. The main motivation of this course is to provide hands on experiences of working with different encryption algorithms, attacking systems exploiting different vulnerabilities and adopting security measures to counteract these vulnerabilities.

Objectives:

- To assist students in developing practical knowledge about different cryptographic algorithms
- To help students to identify different vulnerabilities within a system and to assess its security.
- To assist students in developing secure systems using different cryptographic libraries.
- To facilitate the knowledge of exploiting network vulnerabilities, attacking as well as defending a web application.

Course Contents:

Attacking classic cipher systems, Programming different cryptographic algorithms, Developing secure systems utilizing different cryptographic libraries, Exploiting network vulnerabilities, attacking and defending web applications and Malware analysis.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Design a secure system
CO 2	Demonstrate the knowledge to attack classis cryptographic mechanisms
CO 3	Demonstrate practical knowledge of different types of attacks in practical systems
CO 4	Analyze and identify security vulnerabilities in practical systems

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3		3									
CO 2	3											
CO 3	3											
CO 4				3								

Textbooks

1. Introduction to Computer Security by Michae lT. Goodrich and Roberto Tamassia
2. Computer Security: Principles and Practice by William Stallings Lawrie Brown, 4th Edition
3. Introduction to Computer Security by Matt Bishop

Course Title: Computer Networking	Credits: 3.0
Course Code: 0612 4165	**Contact hours:** 3 hours/week

Rationale:

The aim of this course is to introduce key concepts and principles of computer networks to provide a solid understanding of the technologies that support modern networked computer systems. The course will use a top-down approach to study the Internet and its protocol stack. Instances of architecture, protocol, application - examples will include email, web and media-streaming. It will cover communications services (e.g., TCP/IP) required to support such network applications. The implementation and deployment of communications services in wired and wireless LAN environments will be followed by a discussion of issues of network-security and network-management. Throughout the course, the Internet's architecture and protocols will be used as the primary examples to illustrate the fundamental principles of computer networking.

Objective:

- To provide basic knowledge about various network technologies and techniques
- To facilitate idea about the importance of layering, and the OSI reference model
- To provide knowledge of understanding of the design and operation of an IP network, such as the Internet, and explain the purpose and function of its various components
- To make them understand the general principles behind addressing, routing, reliable transmission and other stateful protocols as well as specific examples of each
- To make them be able to describe the issues in connecting heterogeneous networks

Course Content:

Introduction: Introduction to Computer Networks, Network Goals, Applications of Networks, Network Structure, Network Architectures, The OSI Reference Model, Data Transmission in the OSI Model, OSI Terminology, The ARPANET. **Local Area Network:** LAN Technology - Architecture, Topology. **Wired LANs:** Ethernet and Fast-Ethernet, Token Ring, FDDI. **Wireless LANs:** IEEE 802.11, Bluetooth. Backbone Networks, Virtual LANs. **Wide Area Network:** SONET, Virtual Circuit Networks - Frame Relay, ATM and ATM LANs. **Network Layer:** Logical Addressing.**Internet Protocol:** Internetworking, Routing Protocol, IPv4 and IPv6. **Address Mapping, Error Reporting and Multicasting:** ICMP, IGMP, ICMv6. Delivery, Forwarding and Routing. **Transport Layer:** Process-to-Process delivery, Transport Services, Protocol mechanisms, TCP, UDP, SCTP, Congestion and QoS. **Application Layer:** Domain Name System, Abstract Syntax Notation One (ASN.1), Network Management - SNMPv2, Electronic mail - SMTP and MIME, Uniform Resource Locator (URL) and Universal Resource Identifier (URI), Hypertext Transfer Protocol (HTTP). **Wireless and Mobile Networking**: Wireless Networking: Issues and Trends, Wireless Physical Layer Concepts, Wireless Cellular Networks, Mobile IP - IPv4, IPv6, TCP over Wireless, Ad Hoc Networks: Issues and Routing, Wireless Sensor Networks, Wireless Mesh and Multi-Hop Relay Networks, Wireless Network Security, Energy Management in Ad Hoc Wireless Networks.

Course Learning Outcome: After the successful completion of the course, the student will be able to-

CO 1	**Explain**the architecture of a computer network and **describe** how each device in a network communicates with each other
CO 2	**Identify**the basic network protocols in each layer of a TCP/IP stack and **compare** with its counterpart OSI layer
CO 3	**Compare** different LAN technologies and their basic principles
CO 4	**Explain** the use of subnetting and use the technique to divide a large network into smaller logical subnetworks

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3												3
CO 4			3			3						

Textbook:

1. Data Communications and Networking – Behrouz A. Forouzan
2. Computer networks – A. S. Tanenbaum, Addison-Wesley.
3. Computer Networking: a Top-down Approach – James F. Kurose, Keith W. Ross
4. Computer Networks: A Systems Approach – Peterson and Davie.

Course Title: Computer Networking Lab	**Credits:** 1.5
Course Code: 0612 4166	**Contact hours:** 3 hours/week

Rationale:
The aim of this lab course is to provide practical knowledge in computer networks. Students will learn subnetting and will design a network using Packet Tracer and analysis the behavior of TCP/IP layers. The students will gain practical knowledge of configuring Switch, Router, DHCP, FTP servers.

Objective:
- To help students designing and implementing VLSM addressing schemes in a heterogeneous computer network using Packet Tracer
- To make them capable to configure Switch, Router and other end devices
- To make them capable to configure DHCP, SMTP and FTP servers
- To make them capable to design and implement a system using socket programming

Course Content:
Subnetting and designing a network using Packet Tracer. Analysis of the TCP/IP behavior. Exploring several aspects of different Application layer protocols such as HTTP and DNS. Packet analysis. Server configuration: DHCP, SMTP, FTP, Web Switch and Router Configuration. Socket Programming.

Course Learning Outcome:
After the successful completion of the course, the student will be able to

CO 1	**Design** and **implement** a heterogeneous computer network
CO 2	**Analyze** the behavior of different Application and Transport layer protocols
CO 3	**Configure** switch, router and end devices in a networkand test connectivity
CO 4	**Analyze** and **implement** IP addressing and subnetting of IP networks

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1			3									3
CO 2		3		3								
CO 3					3							
CO 4			3	3								

Textbook:
1. Data Communications and Networking – Behrouz A. Forouzan
2. Computer networks – A. S. Tanenbaum, Addison-Wesley.
3. Computer Networking: a Top-down Approach – James F. Kurose, Keith W. Ross
4. Computer Networks: A Systems Approach – Peterson and Davie.

Course Title: Software Project Management	**Credits:** 2.0
Course Code: 0613 4142	**Contact hours:** 2 hours/week

Rationale:
This course attempts to enlighten students on how software projects are managed in the current software industry and what tools are used. Many of the students who graduated from CSE will join the industry. Eventually they will be promoted to become team-lead or project manager. So, it is imperative that students develop the basic skills on how to properly manage a project (planning, scheduling, resource allocation, execution, tracking and delivery of software projects).

Objectives:
- To provide knowledge how to put together the blueprint of the entire project from ideation to fruition
- To give students an insight on how software are managed in the current industry and what tools and standards are used to manage and evaluate these projects
- To help students develop managerial skill-set such as cost estimation, risk management, configuration management, proper use of available resources etc.

- To enhance leadership skills among students so that they can assemble and lead the project team.

Course Contents:
Planning and managing of software development projects. Software process models. ISO 9000, SEI's Capability Maturity Model, continuous process improvement. Planning, scheduling, tracking, cost estimation, risk management, configuration management.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Understand how to put together the blueprint of the entire project from ideation to fruition.
CO 2	Identify what tools and standards are used to manage and evaluate a projects
CO 3	Develop managerial skill-set such as cost estimation, risk management, configuration management, proper use of available resources etc.
CO 4	Enhance leadership skills among students so that they can assemble and lead the project team

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3				2				1			
CO 2			3						2	1		
CO 3					3				2			
CO 4									2		3	

Textbook
1. Software Project Management for Dummies by Joseph Phillips and Teresa Luckey

2. Introduction to Software Project Management by Book by Adolfo Villafiorita

Course Title: Final Year Project II	**Credits:** 4.0
Course Code: 0610 4250	**Contact hours:** 8 hours/week

Rationale:
This course is intended to facilitate students continue their hands-on experience obtained in the course 0610 4150 on a contemporary topic and advance the computing solution through design, development and validation.

Objectives:
- To facilitate students apply theoretical concepts in practical settings.
- To help students develop problem-solving skills.
- To help students develop communication and teamwork skills.
- To help students develop project management skills.
- To help students promote an understanding of professional and ethical responsibilities.

Course Contents:
Project work based on the core courses studied in the previous semesters.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Identify a real-life problem that can be translated to an engineering and/or computing solution through design, development and validation.
CO2	Identify outcomes and functional requirements of the proposed solution considering software and/or hardware specification and standards
CO3	Identify sub-components of a complex problem, prepare timeline and appropriate budget using the project management skill
CO4	Analyze, design, build, and evaluate engineering/computing system/subsystem with given specifications and requirements
CO5	Identify and validate the impact of environmental considerations and the sustainability of a system/subsystem of a complete project

CO6	Assess professional, ethical, and social impacts and responsibilities of the design project.
CO7	Function effectively in a multi-disciplinary team
CO8	Use modern analysis and design tools in the process of designing and validating of a system and subsystem
CO9	Present design project results through written technical documents and oral presentations

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2	3	2										
CO 3											3	
CO 4			3	3								
CO 5							3					
CO 6						3		3				
CO 7									3			
CO 8					3							
CO 9										3		

Course Title: Thesis II	**Credits:** 4.0
Course Code: 0610 4280	**Contact hours:** 8 hours/week

Rationale:
This course is intended to advance students' research experience developed in the course 0610 4180 in identifying, designing, conducting, and reporting research works ona contemporary topic.

Objectives:
- To encourage students foster creativity and innovation
- To help students conduct literature review on current research
- To help students learn basic idea about research methodologies
- To help develop knowledge about research proposal
- To help develop skill of presentation
- To promote an understanding of ethical and professional considerations in research.

Course Contents:
Research work based on the core courses studied in the previous semesters.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Identify and formulate a research question through literature review on state-of-the-arts.
CO2	Competence in using appropriate research methods and techniques.
CO3	Prepare timeline and appropriate budget using the research project management skill.
CO4	Collect, analyze, and interpret data.
CO 5	Identify and validate the impact of environmental considerations and the sustainability of a system/subsystem of a research project
CO 6	Assess professional, ethical, and social impacts and responsibilities of the research project.
CO 7	Ability to collaborate with others and work effectively in a team.
CO 8	Use modern analysis and design tools in the process of developing and validating of a system
CO 9	Present research results through written technical documents and oral presentations.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2	3											
CO 3											3	
CO 4			3	3								
CO 5							3					
CO 6						3		3				
CO 7									3			
CO 8					3							
CO 9										3		

Course Title: Comprehensive Viva Voce	**Credits:** 1.5
Course Code: 0610 4290	**Contact hours:** 3 hours/week

Rationale:
This course objects to prepare students for their upcoming real life interviews.

Objectives:
- To make students get mentally prepared for real life interviews
- To make them recall all the important and fundamental knowledge they have acquired during the full undergrad session

Course Contents:
Viva based on studied major courses.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Present skills on facing verbal sessions.
CO 2	Recall fundamental information they acquired in their undergrad life
CO 3	Argue logically and defend their answer

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1												3
CO 2	3	2										
CO 3											3	

Course Title: Engineering Ethics And Cyber Law	**Credits:** 2.0
Course Code: 0688 4247	**Contact hours:** 2 hours/week

Rationale:
This course consists of a sustained study of ethical and legal issues that arise in relation to employment in the public and private sectors, including allocation of resources, corporate and social responsibility, relationships, and discrimination. A main focus of this course will be on the ethical and legal standards governing information technology. New technology creates ethical challenges for individuals around the globe and applies to most persons regardless of whether they are employed in the information technology field or a more traditional occupation. The study of Cyber Ethics provides a framework for making ethical decisions that professionals are likely to encounter in the workplace. This course will not only focus on ethics but on the legal, economic, social, cultural, and global impacts of decisions that are made in the context of professional occupations.

Objectives:
- To make students explore and understand ethics and boundaries of morality and technology.
- To help them understand, explore, and acquire a critical understanding of cyber law.
- To facilitate necessary knowledge about different rules for legal bindings.

Course Contents:
Ethics: Introduction. Meta Ethics: Objectivism and Relativism, Non-naturalism, Cognitivism and Non-Cognitivism, The epistemic problem for cognitivism, Moral relativism, Cross-cultural differences and similarities, Different Psychological Issues in Meta-ethics: Egoism and Altruism,

Emotion and Reason, Male and Female morality. Normative Ethics: Goodness, Rightness, Consequentialism, Utilitarianism. Applied Ethics: Business Ethics, Environmental Ethics and Social Ethics, Computer and Information Ethics. Developing the ethical analysis skills and professional values. **Cyber Law: Module I: Introduction:** Computers, Internet and their Impacts in Society; Need for Cyber Law in Social and International Perspectives; Overview of Cyber Law, Cyberspace; Building blocks of CyberSpace; Cyber Jurisprudence at International and National Level; Jurisdictional Aspects in Cyber Law. **Module II: Cyber Crimes & Legal Framework:** Cyber Crimes against Individuals, Institution and State; Hacking; Digital Forgery; Cyber Stalking/Harassment; Cyber Pornography; Identity Theft & Fraud; Cyber terrorism; Cyber Defamation; Different offences under ICT Act, 2006. **Module III: Intellectual Property Issues in CyberSpace:** Interface with Copyright Law; Interface with Patent Law; Trademarks & Domain Names Related issues. **Module IV: E Commerce:** Concept; E-commerce-Salient Features; Online approaches like B2B, B2C & C2C; Online contracts; Click Wrap Contracts; Applicability of Contract Act, 1872. **Module V:** Cyber Tribunal: Establishment of Cyber Tribunal, Trial Procedure of Cyber Tribunal, Bail Rules, Time Limit, Power of Investigation etc.; Cyber Appellate Tribunal: Establishment of Cyber Appellate Tribunal, Procedure and Power Cyber Appellate Tribunal, Appeal Procedure in case of not establishing Cyber Appellate Tribunal.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Apply diverse viewpoints to ethical dilemmas in the information technology field and recommend appropriate actions
CO2	Identify and analyze statutory, regulatory, constitutional, and organizational laws that affect the information technology professional
CO3	Locate and apply case law and common law to current legal dilemmas in the technology field
CO 4	Distinguish enforceable contracts from non-enforceable contracts

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO 1								3			
CO 2							3				
CO 3						3					
CO 4											

Textbook

1. Engineering Ethics (Esource) 4th Edition by Charles Fleddermann, Pearson;

Course Title: Computer Graphics and Multimedia System	**Credits:** 3.0
Course Code: 0613 4231	**Contact hours:** 3 hours/week

Rationale:

In many engineering applications (e.g. automotive, aerospace, medical), the ability to quickly visualize newly designed shapes is indispensable. Using computer graphics, designers can interactively view and modify models of their shapes using a computer. Therefore, a student who is willing to build his/her career in modeling and visualizing the data from imaging this course will help them to learn the fundamentals and tools used to create and manipulate digital graphics.

Objectives:

- To provide knowledge on the basic elements and skills involved in the creation of computer graphics
- To help them to learn how to apply computer graphics skills and capacities to enhance published content
- To facilitate knowledge about how to model and visualize different products, buildings and cars etc. and visualize data from medical imaging such as CT scans
- To help them learn about the connection between computer graphics capacities and skills and workplace career and professional opportunities

Course Contents:

Computer Graphics Programming: OpenGL. **Raster Graphics**: Line Drawing, Anti-aliasing, Polygon Filling Algorithms. **Camera Analogy**: Viewing, Windowing, Clipping. **Projective Transformation (Ray-tracing):** Orthogonal Projection, Perspective Projection. **Vector:** Normal Vector, View Vector. **Matrix**: 2D and 3D Rotation and Translation Matrix. **Hidden Surface Removal**: z-buffering. **Lighting and Surface Property:** Diffused Light, Ambient Light, Specular Light, Lighting Models for reflection. **Shading**: Flat Shading, Lambert Shading, Phong Shading. **Texture Mapping**: Texture Fundamentals. **Animation:** Real time animation

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Select and analyze fundamentals such as digital image representation, color perception, image formation, and image processing
CO2	Apply algorithms related to hidden surface removal that includes but are not limited to the Z-buffer algorithm and the Painter's algorithm
CO3	Elaborate the algorithmic and mathematical tools that are used to create a variety of digital images and effects
CO4	Demonstrate three main subjects within computer graphics, modeling, rendering, and animation

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3	3									
CO 2	3	3	3									
CO 3	3	3	3	3	3							
CO 4	3	3	3	3	3							

Textbook

1. Theory and Problems of Computer Graphics (3rd Edition) – Zhigang Xiang, Roy A. Plastock; McGraw Hill (2000).
2. Computer Graphics C Version (3rd Edition) – Donald Hearn, M. Pauline Baker; Pearson Prentice Hall (2004).
3. Computer Graphics Principle and Practice (3rd Edition) – Donald Hearn, M. Pauline Baker; Addison-Wesley Professional (2013).

Course Title: Computer Graphics and Multimedia System Lab	**Credits:** 1.5
Course Code: 0613 4232	**Contact hours:** 3 hours/week

Rationale:
This course motivates to develop and modify 2D and 3D visualization and transformation of any geometric object by using graphics library as well as working with texturing, lighting and coloring of such objects to develop different types of digital images with various effects.

Objectives:

- To learn basic concepts of 2D, 3D and animation graphics project using OpenGL graphics library
- To understand graphics programming and familiar with image manipulation, enhancement.
- To familiarize with 3D graphical scenes using open graphics library suits and learn to create animations and multimedia presentation/Game/Project.
- To develop 3D games and animation using different software like blender, unity etc.

Course Contents:
Computer Graphics Programming: OpenGL, **Scan Conversion:** Implementation of Algorithms for drawing 2D Primitives – Line (DDA, Bresenham), Circle (Bresenham ,Midpoint), **Region Filling**: Scan line algorithm, **Transformation:** 2D Geometric transformations – Translation, Rotation, Scaling, Reflection, Shear Window-Viewport, Composite 2D Transformations, 3D Transformations - Translation, Rotation, Scaling, **Clipping**: Line Clipping, polygon clipping, **Projections**: 3D Projections – Parallel, Perspective, **Animation**: 2D Animation, Interactive animation using any authoring tool

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Learn and apply the graphics library OpenGL
CO 2	Draw basic geometric shapes (Points, Lines, Circles) using drawing algorithms.
CO 3	Apply geometrical transformations on graphical problem solving.
CO 4	Develop skill to generate computer graphics animation software.
CO 5	Demonstrate 2D and 3D graphics processing techniques. (transformation, viewing, clipping)

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1					3		3					
CO 2						3						
CO 3						3						
CO 4								3				
CO 5									3	3		

Textbook

1. OpenGL Programming Guide: The Official Guide to Learning OpenGL (8th Edition)- Dave Shreiner, Graham Sellers, John Kessenich and Bill Licea-Kane; Addison Wesley Professional (2013)
2. Theory and Problems of Computer Graphics (3rd Edition) – Zhigang Xiang, Roy A. Plastock; McGraw Hill (2000).
3. Computer Graphics C Version (3rd Edition) – Donald Hearn, M. Pauline Baker; Pearson Prentice Hall (2004).
4. Computer Graphics Principle and Practice (3rd Edition) – Donald Hearn, M. Pauline Baker; Addison-Wesley Professional (2013).

OPTIONAL COURSES

Course Title: Security Engineering	**Credits:** 3.0
Course Code: 0612 4261	**Contact hours:** 3 hours/week

Rationale:
This course provides an introduction to the new area of Security Engineering, and provides examples drawn from recent research. This is a multi-disciplinary field combining technical aspects of Applied Cryptography, Computer Security, Computer Engineering, and Networking as well as issues from Psychology, Policy and Economics.

Objectives:

- To facilitate the application-oriented knowledge of different aspects of computer security.
- To assist students in developing an understanding regarding the engineering aspects of security.
- To help students conceptualize basic theories of authentication, authorization, privacy, security usability, security protocols and blockchain engineering.
- To assist students in developing basic knowledge about security usability, secure systems development, security and privacy by design, security economics.
- To provide the engineering knowledge of blockchain systems

Course Contents:
Security Protocols: The need for security protocols, developing security protocol, verification of security. **Identity & Authentication:** Identification and Authentication definitions, authentication methods. **Authorization:** Authorization concepts, authorization models, Role Based Access Control (RBAC). **Trust Management:** concepts, managing trust online, trust management models. **Application of Economics in Security**: Information economics,Game Theory, The economics of security and dependability. **Secure Systems Development:** Managing a security project, Methodology, Security Engineering, Risk management. **System Evaluation and Assurance**: System assurance, Security Evaluation. **Secure Monitoring**:Securing logs, monitoring activities within a system. Privacy Issues: Privacy enhancing technologies, anonymous emails, anonymous web

browsing, TOR.**Security and privacy by design paradigm:** Concepts and methodologies**.Security Usability**: attacks based on psychology, social engineering attacks, CAPTCHA, balancing security and usability.**Blockchain engineering:** Integrating security protocols with blockchain applications.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	Design security protocols for different applications
CO 2	Explain the theoretical foundation of access control models
CO 3	Demonstrate the practical application of authentication and authorization for a secure system
CO 4	Analyze and identify security requirements for secure software development
CO 5	Demonstrate knowledge about different tools that are used to secure different systems
CO 6	Evaluate risks and assurance factors within a system
CO 7	Evaluate the importance of human resources within an organization to maintain its security practices
CO 8	Analyzeprivacy issues within a system

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3		3									3
CO 2	3											3
CO 3		3			3							
CO 4		3		3								
CO 5	3											
CO 6		3		3								
CO 7						3						
CO8		3		3								

Textbooks

1. Security Engineering: A Guide to Building Dependable Distributed Systems – Ross Anderson
2. Cryptography Engineering: Design Principles and Practical Applications– Niels Ferguson, Bruce Schneier, Tadayoshi Kohno

Course Title: Security Engineering Lab	**Credits:** 1.5
Course Code: 0612 4262	**Contact hours:** 3 hours/week

Rationale

In this course, these students will carry out a number of hands-onlab works based on concepts gained in its counterpart theory course, CSE 4163. The main motivation of this course is to provide hands on experiences of working with different security protocols, security tools, secure software development, secure project management, risk management, privacy enhancing technologies and blockchain development.

Course Objectives:

- To assist students in developing practical knowledge about developing secure systems
- To assist students in designing cryptographic protocols and verifying them using a verification tool
- To help students to manage secure software development
- To assist students in utilizing privacy enhancing technologies to ensure the privacy of the users
- To facilitate the knowledge so that a student can integrate the effective authentication and authorization models in a system

Course Contents:

Writing and verification of security protocols, authentication tools, authorization tools, risk management, secure software management, designing a usable secure system, system monitoring mechanisms, evaluating the security of a system, privacy enhancing technologies and blockchain system development.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	**Implement** security protocols
CO 2	**Design** secure systems
CO 3	**Apply** authentication and authorization mechanisms into any system
CO 4	**Analyze** privacy issues within a system
CO 5	**Demonstrate** knowledge about privacy enhancing technologies
CO 6	**Design** and **implement** blockchain based systems

Mapping of Course Outcomes (COs) to Program Learning Outcomes:

Cos\POs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	3										3
CO 2	3		3									3
CO 3	3											
CO 4		3										
CO 5	3											
CO 6	3											

Textbooks

1. Security Engineering: A Guide to Building Dependable Distributed Systems – Ross Anderson
2. Cryptography Engineering: Design Principles and Practical Applications– Niels Ferguson, Bruce Schneier, Tadayoshi Kohno

Course Title: Advanced Database System	**Credits:** 3.0
Course Code: 0612 4163/4263	**Contact hours:** 3 hours/week

Rationale:

This course motivates to optimize the basic database transactions, query processing, concurrency control and other functions of database systems using advanced features that includes complex data and also assess various database models and designs to contribute to modern database systems.

Objectives:

- To introduce the concepts and implementation schemes in database management systems such as advanced access methods, query processing and optimization, transactions and concurrency control.
- To analyze and evaluate different models and methods of database systems for certain context using complex data and functions.

Course Contents:

Object oriented database; data model, design, languages; object relational database: complex data types, querying with complex data types, design; distributed database: levels of distribution transparency, translation of global queries to fragment queries, optimization of access strategies, management of distributed transactions, concurrency control, reliability, administration; Parallel Database: different types of parallelism, design of parallel database; multimedia database systems basic concepts, design, optimization of access strategies, management of multimedia database systems, reliability; database warehousing/ data mining: basic concepts and algorithms.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Explain and evaluate the fundamental theories and requirements that influence the design of modern database systems.
CO2	Assess and apply database functions and packages suitable for enterprise database development and database management.
CO3	Critically evaluate alternative designs and architectures for databases and data warehouses.
CO 4	Discuss and evaluate methods of storing, managing and interrogating complex data.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1		3										
CO 2			3									
CO 3		3										
CO 4		3										

Textbook

1. Database Systems: The Complete Book (2nd Edition) - Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom; Pearson (2008)
2. Concurrency control and Recovery in Database Systems (1st Edition) - P. a Bernstein and N. Goodman; Addison-Weasley (1987)

Course Title: Advanced Database System Lab	**Credits:** 1.5
Course Code: 0612 4164/4264	**Contact hours:** 3 hours/week

Rationale:

This course motivates to design and develop embedded projects using advanced database functions and query based on advanced database models - object oriented database, distributed database, multimedia database etc. to solve real-life problems.

Objectives:

- To develop embedded projects for different applications using advanced database functions.
- To analyze different security aspects of complex data transactions using different database techniques.

Course Contents:
Object oriented database, Distributed database, Management of distributed transactions, concurrency control, reliability, administration, Management of multimedia database systems, reliability; database ware-housing/data mining: basic concepts and algorithms.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Solve and apply the advanced knowledge in different projects with a commercial relational database system (Oracle).
CO2	Embed security aspects in the developed systems aspects of data transaction.
CO3	Explain the methods of storing, managing and interrogating complex data.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1						3						
CO 2							3					
CO 3												
CO 4											3	

Textbook

1. Database Systems: The Complete Book (2nd Edition) - Hector Garcia-Molina, Jeffrey D. Ullman and Jennifer Widom; Pearson (2008)
2. Concurrency control and Recovery in Database Systems (1st Edition) - P. a Bernstein and N. Goodman; Addison-Weasley (1987)
3. JAVA How to Program (9th Edition) – Paul Deitel, Harvey Deitel; Prentice Hall (2011)
4. Microsoft C# Professional Projects (1st Edition) - Geetanjali Arora, B. Aiaswamy, Nitin Pandey; Course Technology PTR (2002)
5. PHP: The Complete Reference (1st Edition) - Steven Holzner; McGraw Hill Education(2007)

Course Title: Cloud Computing	**Credits:** 3.0
Course Code: 0612 4265	**Contact hours:** 3 hours/week

Rationale:
This course covers the core concepts of cloud computing. Foundational knowledge required for understanding cloud computing from a business perspective as also for becoming a cloud practitioner is covered. Topics include the definition and essential characteristics of cloud computing, its history, the business case for cloud computing, and emerging technology use cases enabled by cloud. Other topics may include a prominent service provider of modern times (e.g. AWS) the services it offers, and look at some case studies of cloud computing across industry verticals.

Objectives:
- To introduce the concepts and implementation schemes in database management systems such as advanced access methods, query processing and optimization, transactions and concurrency control.
- To analyze and evaluate different models and methods of database systems for certain context using complex data and functions.

Course Contents:
This course defines Cloud Computing and establishes a strong working knowledge of the concepts and technologies needed to work effectively with the cloud. The course allows to understand what cloud computing is and how it works. It describes the benefits of cloud computing along with its potential drawbacks. The course enables to determine which cloud is appropriate from a business and technical perspective, to select appropriate cloud providers and to plan and implement a cloud adoption strategy. Formation of the theoretical knowledge and practical skills in practical realization of the benefits of cloud computing in today's business, learning the tools of the technology. The course covers concepts required to build classic (traditional), virtualized, and cloud data center environments. These concepts include compute, storage, networking, desktop and application virtualization.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Identify key components of the cloud computing paradigm.
CO2	Analyze different cloud computing principles to meet the user's Service Level Agreement (SLA) and ensure Quality-of-Service (Qos).
CO3	Create designs of systems leveraging cloud resources to benefit from cloud infrastructure.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Cloud Computing: Principles and Paradigms, 1st Edition, by Rajkumar Buyya, James Broberg , Andrzej M. Goscinski

Course Title: Cloud Computing Lab	**Credits:** 1.5
Course Code: 0612 4266	**Contact hours:** 3 hours/week

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1		3										
CO 2			3									
CO 3					2							

Rationale:

This course is all about the various areas of cloud computing. Students will have to learn about how cloud computing works in various areas, along with the application of many cloud computing techniques on real-world scenarios. This course covers the practical part of the core concepts of cloud computing. Topics include the definition and essential characteristics, its history, the business case for cloud computing, and emerging technology use cases enabled by cloud.

Objectives:

Presentation of cloud computing as a coherent body of ideas and methods to acquaint the student with the basic ideas in the field and their underlying theory. Students will explore this through the use of a simulation framework to test out cloud computing ideas and will be introduced to a real-world cloud platform.

Course Contents:

Introduction to CloudSim simulation framework, using multiple cloudlets and VMs,using multiple hosts, multiple data centers and using different VM scheduler policies,creating a scalable simulation,creating simulation entities dynamically, pausing and resuming simulation,creating simulation entities in run-time using a global manager entity,using a datacenter with network topology,using multiple datacenters with multiple hosts with network topology Introduction to AWS Free Tier,using AWS Free Tier Final Term Examination (Lab test, Viva) Project Presentation

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1		3										
CO 2			3									
CO 3					3							

CO1: Design scalable systems for societal welfare by employing resources provided by cloud service providers.
CO2: Develop sustainable solutions on cloud platforms by utilizing various aspects of cloud infrastructure.
CO3: Adapt cloud computing skills to work individually and contribute to effective team collaboration.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. The Cloud Computing Book:The Future of Computing Explained By Douglas Comer Published July 1, 2021 by Chapman and Hall/CRC

Course Title: Parallel and Distributed Computing	**Credits:** 3.0
Course Code: 0612 4167/4267	**Contact Hours:** 3 hours/week

Rationale:

Distributed computing is the method of making multiple computers work together to solve a common problem. It makes a computer network appear as a powerful single computer that provides large-scale resources to deal with complex challenges.

Objectives:

- Understand different parallel and distributed programming paradigms and algorithms
- Gain practice in implementing and testing solutions using these.

Course Contents:

Introduction: Scope, issues, applications and challenges of Parallel and Distributed Computing; Parallel and Distributed Architectures. **Parallel Computing:** Parallel Performance; Shared Memory and Threads; Parallel Algorithms; OpenMP; Scalable Algorithms; Message Passing;MPI; Grid Computing. **Distributed Computing**: Distributed Systems; MapReduce; Clusters; Distributed Coordination; Distributed Consensus; Distributed File Systems (DFS); Distributed Shared Memory; Distributed Transactions and

Replication; Applications of Distributed Computing in Security. **Cloud Computing**: Cloud Architectural Framework; Cloud Deployment Models: Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS); Cloud Economics; Service Level Agreements (SLA); Security and Privacy issues in Cloud Computing.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

CO 1: Understand the requirements for programming parallel systems and how they can be used to facilitate the programming of concurrent systems.

CO 2: To learn and apply knowledge of parallel and distributed computing techniques and methodologies

CO 3: To gain experience in the design, development, and performance analysis of parallel and distributed applications

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					2						
CO 2	3	2		3	1							
CO 3			3				3		2		3	1

Textbook

1. Distributed Systems: Concepts and Designs - George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair

Course Title: Parallel and Distributed Computing Lab	**Credits:** 1.5
Course Code: 0612 4168/4268	**Contact Hours:** 3 hours/week

Rationale:

Distributed computing is the method of making multiple computers work together to solve a common problem. It makes a computer network appear as a powerful single computer that provides large-scale resources to deal with complex challenges.

Objectives:

- Apply different parallel and distributed programming paradigms and algorithms
- Gain practice in implementing and testing solutions using these.

Course Contents:

Introduction: Scope, issues, applications and challenges of Parallel and Distributed Computing; Parallel and Distributed Architectures. **Parallel Computing:** Parallel Performance; Shared Memory and Threads; Parallel Algorithms; OpenMP; Scalable Algorithms; Message Passing;MPI; Grid Computing. **Distributed Computing**: Distributed Systems; MapReduce; Clusters; Distributed Coordination; Distributed Consensus; Distributed File Systems (DFS); Distributed Shared Memory; Distributed Transactions and Replication; Applications of Distributed Computing in Security. **Cloud Computing**: Cloud Architectural Framework; Cloud Deployment Models: Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS); Cloud Economics; Service Level Agreements (SLA); Security and Privacy issues in Cloud Computing.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

CO1: Understand the requirements for programming parallel systems and how they can be used to facilitate the programming of concurrent systems.

CO2: To learn and apply knowledge of parallel and distributed computing techniques and methodologies

CO3: To gain experience in the design, development, and performance analysis of parallel and distributed applications

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					2						
CO 2	3	2		3	1							
CO 3			3				3		2		3	1

Textbook

1. Distributed Systems: Concepts and Designs - George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair

Course Title: Digital Image Processing	**Credits:** 3.0
Course Code: 0613 4131	**Contact hours:** 3 hours/week

Rationale:

This course provides an introduction to basic concepts, methodologies, and algorithms of digital image processing for image analysis and information retrieval.

Objectives:

- To introduce the concepts of image processing and basic analytical methods to be used in image processing.

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3		3							
CO 3			3		3							
CO 4				3								
CO 5			3		3							

- To familiarize students with image enhancement and restoration techniques.
- To explain different image compression techniques.
- To introduce segmentation and morphological processing techniques

Course Contents:

Digital image fundamentals: Fundamentals of Image formation, components of image processing system, image sampling and quantization. **Image enhancement in the spatial domain:** Basic gray-level transformation, histogram processing, arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters. **Image enhancement in the frequency domain:** Introduction to Fourier Transform, Smoothing and Sharpening frequency domain filters, Ideal, Butterworth and Gaussian filters**. Image restoration:** A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the image enhance in frequency domain. **Image Compression:** Need of image compression, image compression models, error-free compression, lossy predictive coding, image compression standards. **Morphological Image Processing:** Preliminaries, dilation, erosion, open and closing, basic morphologic algorithms, The Hit-or-Miss Transformation. **Image Segmentation:** Detection of discontinuous, edge linking and boundary detection, thresholding, Hough Transform Line Detection and Linking, region–based segmentation. **Object Recognition:** Patterns and patterns classes, matching, classifiers.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Explain the fundamentals of digital image and its processing
CO2	Perform image enhancement techniques in spatial and frequency domain
CO3	Elucidate the mathematical modeling of image restoration and compression
CO4	Apply the concept of image segmentation
CO5	Describe object detection and recognition techniques

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Digital Image by R. C. Gonzalez and R.E. Woods (Pearson Prentice Hall)
2. Digital Image Processing Using MATLAB (Third Edition) by Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins
3. Fundamentals of Digital Image Processing by Anil Jain K.

Course Title: Digital Image Processing Lab	**Credits:** 1.5
Course Code: 0613 4132	**Contact hours:** 3 hours/week

Rationale:

This course is designed to provide a practical understanding of the essential properties of digital image processing for image analysis and information retrieval

Objectives:

- To study the image fundamentals and mathematical transforms necessary for image processing.
- To implement the image enhancement techniques

- To apply image restoration procedures in different applications
- To study the image compression procedures
- To familiarize with the image segmentation techniques through examples

Course Contents:
Digital image fundamentals: Fundamentals of Image formation, components of image processing system, image sampling and quantization. **Image enhancement in the spatial domain:** Basic gray-level transformation, histogram processing, arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters. **Image enhancement in the frequency domain:** Introduction to Fourier Transform, Smoothing and Sharpening frequency domain filters, Ideal, Butterworth and Gaussian filters**. Image restoration:** A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the image enhance in frequency domain. **Image Compression:** Need of image compression, image compression models, error-free compression, lossy predictive coding, image compression standards. **Morphological Image Processing:** Preliminaries, dilation, erosion, open and closing, basic morphologic algorithms, The Hit-or-Miss Transformation. **Image Segmentation:** Detection of discontinuous, edge linking and boundary detection, thresholding, Hough Transform Line Detection and Linking, region–based segmentation. **Object Recognition:** Patterns and patterns classes, matching, classifiers.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Design and implement algorithms that perform basic image processing (e.g., noise removal and image enhancement)
CO2	Design and implement algorithms for advanced image analysis (e.g., image compression, image segmentation & image representation)
CO3	Assess the performance of image processing algorithms and systems.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1			3		3							
CO 2			3		3							
CO 3				3								

Textbook
1. Digital Image by R. C. Gonzalez and R.E. Woods (Pearson Prentice Hall)
2. Digital Image Processing Using MATLAB (Third Edition) by Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins
3. Fundamentals of Digital Image Processing by Anil Jain K.

Course Title: Advanced Data Structure and Algorithm	**Credits:** 3.0
Course Code: 0613 4137/4237	**Contact Hours:** 3 Hours/Week

Rationale:
This course introduces to a variety of algorithms, approaches to algorithm design, and how they are effectively applied to solve problems in computer science. This course provides material that is more advanced and, in more depth, than an undergraduate data structures course, with a focus on algorithms and analysis.

Objectives:
- To analyze algorithms and to determine algorithm correctness and time efficiency class.

- To master a variety of advanced abstract data type (ADT) and data structures and their implementations.

Course Contents:
Red-Black Tree, Binary Index Tree, Segment Tree, Range minimum query, lowest common ancestor, k-d Tree, Interval tree, R-tree. Advanced Application of Dynamic Programming and Backtracking. **Advanced String Structure and algorithm:** tree, suffix tree, suffix array, Aho-Corasic. **Computational Geometry:** Line Sweeping algorithms, Binary Space Partition Trees and Painter’s algorithm (other advanced computational geometry). **Optimization of network flow:** Dinic's algorithm, Hungarian algorithm, Min cost max flow, min cut, graph coloring. Genetic algorithm and its different applications, Basic Game theory, Linear programming, Polynomials and Fast Fourier Transform, Encryption and Decryption.

Course Learning Outcomes, CO
After the successful completion of the course, students will be able to:
CO 1: Analyze algorithms and to determine algorithm correctness and time efficiency class.

CO 2: Master a variety of advanced abstract data type (ADT) and data structures and their implementations.

CO 3: Apply and implement learned algorithm design techniques and data structures to solve problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Introduction to Algorithms (Third Edition) - By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein

Course Title: Advanced Data Structure and Algorithm Lab	**Credit:** 1.5
Course Code: 0613 4138/4238	**Contact Hours:** 3 Hours/Week

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					1						
CO 2		3			2							
CO 3			2				3		1			

Rationale:

This course introduces to a variety of algorithms, approaches to algorithm design, and how they are effectively applied to solve problems in computer science. This course provides material that is more advanced and, in more depth, than an undergraduate data structures course, with a focus on algorithms and analysis.

Objectives:

- To analyze algorithms and to determine algorithm correctness and time efficiency class.
- To master a variety of advanced abstract data type (ADT) and data structures and their implementations.

Course Contents:

Red-Black Tree, Binary Index Tree, Segment Tree, Range minimum query, lowest common ancestor, k-d Tree, Interval tree, R-tree. Advanced Application of Dynamic Programming and Backtracking. **Advanced String Structure and algorithm:** tree, suffix tree, suffix array, Aho-Corasic. **Computational Geometry:** Line Sweeping algorithms, Binary Space Partition Trees and Painter's algorithm (other advanced computational geometry). **Optimization of network flow:** Dinic's algorithm, Hungarian algorithm, Min cost max flow, min cut, graph coloring. Genetic algorithm and its different applications, Basic Game theory, Linear programming, Polynomials and Fast Fourier Transform, Encryption and Decryption.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					1						
CO 2		3			2							
CO 3			2				3		1			

CO 4: Analyze algorithms and to determine algorithm correctness and time efficiency class.

CO 5: Master a variety of advanced abstract data type (ADT) and data structures and their implementations.

CO 6: Apply and implement learned algorithm design techniques and data structures to solve problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Introduction to Algorithms (Third Edition) - By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein

Course Title: Natural Language Processing	**Credits**: 3.0
Course Code: 0613 4143/4243	**Contact Hours**: 3 Hours/Week

Rationale:

As momentum for machine learning and artificial intelligence accelerates, natural language processing (NLP) plays a more prominent role in bridging computer and human communication. This course is intended as a theoretical and methodological introduction to the most widely used and effective current techniques, strategies and toolkits for speech and language processing.

Objectives:

- teach students the leading trends and systems in natural language processing.
- make them understand the concepts of morphology, syntax, semantics and pragmatics of the language and that they are able to give the appropriate examples that will illustrate the above mentioned concepts.
- enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

Course Contents:

Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms, The State of the Art, Brief History. **Lexical Computation:** Regular Expressions, Words, Corpora, Text Normalization, Minimum Edit Distance. **N-gram Language Models:** N-grams, Evaluating Language Models, Smoothing. **Naïve Bayes, Text Classification, and Sentiment:** Naïve Bayes Classifiers, Text Classification, Sentiment Analysis, Test Sets and Cross-Validation, Evaluation: Precision, Recall, F-measure. **Logistic Regression:** Classification with Logistic Regression, Gradient Descent, Regularization. **Semantics and Embeddings:** Words and vectors, TF-IDF, PMI, Word2Vec, Embeddings. **Neural Language Models:** Transformers as Language Models, Pretraining Large Language Models, Language Models for Zero Learning. **Sequence Labeling:** Part-of-Speech Tagging, Named Entity Tagging, Conditional Random Fields. **Parsing:** Context-Free Grammars, Treebanks, CKY Parsing, Neural Constituency Parsing, Dependency Parsing. **Computational Semantics:** Logical Representations of Sentence Meaning, Relation and Event Extraction, Time and Temporal Reasoning, Word Senses and WordNet, Semantic Role Labeling, Lexicons, Coreference Resolution, Discourse Coherence. **Phonetics:** Speech Sounds and Phonetic Transcription, Articulatory Phonetics, Prosody, Acoustic Phonetics and Signals. **Applications:** Machine Translation, Question Answering and Information Retrieval, Chatbots and Dialogue Systems, Automatic Speech Recognition and Text-to-Speech.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

CO 7: gain a foundational understanding in natural language processing methods and strategies.

CO 8: evaluate the strengths and weaknesses of various NLP technologies and frameworks as they gain practical experience in the NLP toolkits available.

CO 9: design solutions to real-life problems using NLP-based analytic techniques.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					3						
CO 2	3	3		3	3							
CO 3			3				3		3		3	3

Textbook

1. Speech and Language Processing (3rd Ed.), By Daniel Jurafsky and James H. Martin
2. Natural Language Processing with Python, By Steven Bird, Ewan Klein, Edward Loper

Course Title: Natural Language Processing Lab	**Credits**: 3.0
Course Code: 0613 4144/4244	**Contact Hours:** 3 Hours/Week

Rationale:

As momentum for machine learning and artificial intelligence accelerates, natural language processing (NLP) plays a more prominent role in bridging computer and human communication. This course is intended as a theoretical and methodological introduction to the most widely used and effective current techniques, strategies and toolkits for speech and language processing.

Students who complete this course will They will also learn how to evaluate the strengths and weaknesses of various NLP technologies and frameworks as they gain practical experience in the NLP toolkits available. Students will also learn how to employ literary-historical NLP-based analytic techniques like stylometry, topic modeling, synsetting and named entity recognition in their personal research.

Objectives:

The objectives of this course are to:

- teach students the leading trends and systems in natural language processing.
- make them understand the concepts of morphology, syntax, semantics and pragmatics of the language and that they are able to give the appropriate examples that will illustrate the above mentioned concepts.
- enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

Course Contents:

Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms, The State of the Art, Brief History. **Lexical Computation:** Regular Expressions, Words, Corpora, Text Normalization, Minimum Edit Distance. **N-gram Language Models:** N-grams, Evaluating Language Models, Smoothing. **Naïve Bayes, Text Classification, and Sentiment:** Naïve Bayes Classifiers, Text Classification, Sentiment Analysis, Test Sets and Cross-Validation, Evaluation: Precision, Recall, F-measure. **Logistic Regression:** Classification with Logistic Regression, Gradient Descent, Regularization. **Semantics and Embeddings:** Words and vectors, TF-IDF, PMI, Word2Vec, Embeddings. **Neural Language Models:** Transformers as Language Models, Pretraining Large Language Models, Language Models for Zero Learning. **Sequence Labeling:** Part-of-Speech Tagging, Named Entity Tagging, Conditional Random Fields. **Parsing:** Context-Free Grammars, Treebanks, CKY Parsing, Neural Constituency Parsing, Dependency Parsing. **Computational Semantics:** Logical Representations of Sentence Meaning, Relation and Event Extraction, Time and Temporal Reasoning, Word Senses and WordNet, Semantic Role Labeling, Lexicons, Coreference Resolution, Discourse Coherence. **Phonetics:** Speech Sounds and Phonetic Transcription, Articulatory Phonetics, Prosody, Acoustic Phonetics and Signals. **Applications:** Machine Translation, Question Answering and Information Retrieval, Chatbots and Dialogue Systems, Automatic Speech Recognition and Text-to-Speech.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

CO 1: gain a foundational understanding in natural language processing methods and strategies.

CO 2: evaluate the strengths and weaknesses of various NLP technologies and frameworks as they gain practical experience in the NLP toolkits available.

CO 3: design solutions to real-life problems using NLP-based analytic techniques.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					3						
CO 2	3	3		3	3							
CO 3			3				3		3		3	3

Textbook

1. Speech and Language Processing (3rd Ed.), By Daniel Jurafsky and James H. Martin
2. Natural Language Processing with Python, By Steven Bird, Ewan Klein, Edward Loper

Course Title: Compiler Construction	**Credits:** 3.0
Course Code: 0613 4147/4247	**Contact hours:** 3 hours/week

Rationale:

This course is designed to provide the idea of internal mechanisms of compiler technologies. Students will be able to know the systematic way of designing a compiler from scratch up to machine level. It will increase the ability to develop any system level evolution.

Objectives:

- To provide knowledge about the structure of compilers;
- To acquaint basic analysis phases of compiler construction such as lexical analysis and parsing, semantic analysis.
- To provide about the fundamental synthesis phases of students like as intermediate representation for machine independent compiler design, code generation and optimization.
- To acquaint with basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines.
- To develop skill to the students about the implementation of a simple compiler using a software engineering approach.

Course Contents:

Introduction to compilers: Introductory concepts about the structure compilers, applications, phases of a compiler,symbol tables.**Lexical analysis**: Role of the lexical analyzer, input buffering, token specification, recognition of tokens, **Syntax Analysis:**Parser and its role, context free grammars, top-down parsing, bottom-up parsing.**Syntax-directed translation**: Syntax-directed definitions, construction of syntax trees, top-down translation and application. **Intermediate representation:** Variants, three-address code, type expressions, static and dynamic checking of types, error recovery. **Run-time organization**: Run-time storage organization, storage strategies. **Code generation**:Issues, target language, address allocation and optimization. **Code optimization**: Basic concepts of code optimization, principal sources of optimization. **Features of some common compilers:** Characteristic features of C, Pascal and Fortran compilers.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Recognize the basic structure and features of compilers.
CO 2	Interpretthe analysis phase of a compiler
CO 3	Describethe basic data structures used in compiler construction.

CO 4	Compose the machine code and optimize the codes as a final phase of a compiler.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook

1. Compilers: Principles, Techniques, and Tools – Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman.
2. Engineering a Compiler - Keith Cooper, Linda Torczon.
3. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, Thompson Learning, 2003
4. J.P. Bennet, “Introduction to Compiler Techniques”, Second Edition, Tata McGraw-Hill, 2003.

Course Title: Compiler Construction Lab	**Credits:** 1.5
Course Code: 0613 4148/4248	**Contact hours:** 3 hours/week

Rationale:

This course is designed to provide the way of compiler development process and practices. Students will be able to do this by using different systematic way of software development techniques. It will increase the skill to use efficient tools for developing any system level software.

Objectives:

- To provide the students hands on practice about the structural development of compilers.
- To familiarize basic implementation techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, intermediate code generation, optimization, and code generation;
- To make them able to employ the basic data structures in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines.
- To help to design and implement a simple compiler using a software engineering approach.

Course Contents:

Compiler construction lab work is designed to accomplish hands-on experience of structural designand testing so that students can develop the software following the demonstration. This lab includes the scanner and parser generator tools such as Flex, JFlex, CUP, Yacc, Bison etc. Hands-on practices are conducted on three parts.

Part1: Use the tools to conduct lexical analysis on the input source program and generate the set of tokens.

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3							3			
CO 3	3											
CO 4				3	3				3			

Part 2: Parse the source program taking the tokens generated in part 1 using the above tools.

Part 3: Use an assembler to generate machine code and then use the code optimization scheme.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Identify the tokens and implement a parser
CO 2	Compose the machine code and optimize the codes as a final phase of a compiler.
CO 3	Evaluate and testing the implementation using software engineering approach.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3		3		3				3			
CO 2		3	3		3				3			
CO 3						3						3

Textbook

1. Andrew W. Appel., “Modern Compiler Implementation in Java/C”.
2. Henk Alblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001.
3. Allen I. Holub “Compiler Design in C”, Prentice Hall of India, 2003.
4. C. N. Fischer and R. J. LeBlanc, “Crafting a compiler with C”, Benjamin Cummings, 2003.

Course Title: Deep Learning	**Credits:** 3.0
Course Code: 0619 4193/4293	**Contact Hours:** 3 hours/week

Rationale:

Most of the advances in artificial intelligence in the last few years have come from deep learning. Neural networks are behind some of the most disruptive and awe-inspiring applications, including autonomous cars, chatbots, image recognition, and robotics. And, with the latest developments in generative AI, the field of deep learning is under more scrutiny than ever before.

Objectives:
The objectives of this course are to:
- make the students understand the theoretical basis underlying neural networks and deep learning.
- develop skills for implementation of neural components as well as applying deep learning on real-world datasets using modern deep learning packages.

Course Contents:
Introduction: The Road to Deep Learning, The Essence of Deep Learning. **Preliminaries:** Data Manipulation and Preprocessing, Linear Algebra, Calculus, Automatic Differentiation, Probability and Statistics. **Linear Neural Networks for Regression:** Linear Regression, Object-Oriented Design, Synthetic Regression Data, Linear Regression Implementation, Generalization, Weight Decay. **Linear Neural Networks for Classification:** Softmax Regression, The Base Classification Model, Softmax Regression Implementation, Generalization in Classification, Environment and Distribution Shift. **Multilayer Perceptrons:** Implementation of Multilayer Perceptrons, Forward Propagation, Backward Propagation, and Computational Graphs, Numerical Stability and Initialization, Generalization in Deep Learning, Dropout. **Convolutional Neural Networks:** Fully Connected Layers to Convolutions, Convolutions for Images, Padding and Stride, Multiple Input and Multiple Output Channels, Pooling, Convolutional Neural Networks (LeNet), Deep Convolutional Neural Networks (AlexNet), Networks Using Blocks (VGG), Network in Network (NiN), Multi-Branch Networks (GoogLeNet), Batch Normalization, Residual Networks (ResNet) and ResNeXt, Densely Connected Networks (DenseNet), Designing Convolution Network Architectures. **Recurrent Neural Networks:** Working with Sequences, Converting Raw Text into Sequence Data, Language Models, Recurrent Neural Network Implementation, Backpropagation Through Time, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Deep Recurrent Neural Networks, Bidirectional Recurrent Neural Networks, The Encoder-Decoder Architecture, Beam Search. **Attention Mechanisms and Transformer:** Attention Pooling by Similarity, Attention Scoring Functions, The Bahdanau Attention Mechanism, Multi-Head Attention, Self-Attention and Positional Encoding, The Transformer Architecture, Large-Scale Pretraining with Transformers. **Optimization Algorithms:** Optimization and Deep Learning, Convexity, Gradient Descent, Stochastic Gradient Descent, Minibatch Stochastic Gradient Descent, Momentum, Adagrad, RMSProp, Adadelta, Adam, Learning Rate Scheduling. **Hyperparameter Optimization:** Asynchronous Random Search, Multi-Fidelity Hyperparameter Optimization, Asynchronous Successive Halving. **Generative Adversarial Networks:** Deep Convolutional Generative Adversarial Networks. **Applications:** Computer Vision, Natural Language Processing, Recommender Systems.

Course Learning Outcomes, CO
After the successful completion of the course, students will be able to:
- **CO 1:** explain the basic principles behind neural networks and deep learning
- **CO 2:** compare modeling aspects of various neural network architectures
- **CO 3:** implement simple neural network algorithms
- **CO 4:** apply and evaluate deep learning on real datasets
- **CO 5:** provide successful examples how deep learning can be used in different contexts in the society

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					2						
CO 2	3	3		3								
CO 3			3		3		2		2			
CO 4							2		3		2	
CO 5						3	2	2				3

Textbook:
1. Dive into Deep Learning, By Aston Zhang, Zachary C. Lipton, Muli, and Alexander J. Smola
2. Deep Learning with Python, By Francois Chollet

Course Title: Deep Learning Lab	**Credits:** 1.5
Course Code: 0619 4194/4294	**Contact Hours:** 3 hours/week

Rationale:
Most of the advances in artificial intelligence in the last few years have come from deep learning. Neural networks are behind some of the most disruptive and awe-inspiring applications, including autonomous cars, chatbots, image

recognition, and robotics. And, with the latest developments in generative AI, the field of deep learning is under more scrutiny than ever before.

Objectives:

The objectives of this course are to:

- make the students understand the theoretical basis underlying neural networks and deep learning.
- develop skills for implementation of neural components as well as applying deep learning on real-world datasets using modern deep learning packages.

Course Content:

Laboratory works based on theory classes.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

CO 1: explain the basic principles behind neural networks and deep learning

CO 2: compare modeling aspects of various neural network architectures

CO 3: implement simple neural network algorithms

CO 4: apply and evaluate deep learning on real datasets

CO 5: provide successful examples how deep learning can be used in different contexts in the society

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3					2						
CO 2	3	3		3								
CO 3			3		3		2		2			
CO 4							2		3		2	
CO 5						3	2	2				3

Textbook:

1. Dive into Deep Learning, By Aston Zhang, Zachary C. Lipton, Muli, and Alexander J. Smola
2. Deep Learning with Python, By Francois Chollet

Course Title: Human Computer Interaction	**Credits:** 3.0
Course Code: 0688 4195/4295	**Contact Hours:** 3 hours/week

Rationale:

Motivates to enrich socio-technological discipline to bring the power of computers and communications systems to people in ways and forms that are both accessible interactively and useful in our working, learning, communicating, and recreational lives.

Objectives:

- To specify fundamental human and computational abilities and constraints and design requirements accordingly by using suitable HCI techniques.
- To design and evaluate the quality of a user interface using various methods to design a usable and useful interface.

Course Contents:

Introduction to Human-Computer Interaction. Task-centered system design: User-centered design and prototyping: methods for involving the user, prototyping, low and medium fidelity. prototypes, Wizard of Oz examples. Methods for evaluation of interfaces with users: goals of evaluation, approaches, ethics, introspection, extracting the conceptual model, direct observation, constructive interaction, interviews and questionnaires, continuous evaluation via user feedback and field studies, choosing an evaluation method. Psychology of everyday things: psychopathology of everyday things, examples, concepts for designing everyday things. Beyond screen design: characteristics of good representations, information visualization, Tufte's guidelines, visual variables, metaphors, direct manipulation. Graphical screen design: graphical design concepts, components of visible language, graphical design by grids. Design principles and usability heuristics: design principles, principles to support usability, golden rules and heuristics, HCI patterns. Semiotic in HCI. HCI design standards: process-oriented standards, product oriented standards, strengths and limitations of HCI Standards. Past and future of HCI: the past, present and future, perceptual interfaces, context-awareness and perception.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Describe the basics of human and computational abilities and limitations.
CO2	Discuss basic theories, tools and techniques in HCI.
CO3	Compare, classify and explain the fundamental aspects of designing and evaluating interfaces

CO4	Practice a variety of simple methods for evaluating the quality of a user interface.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook:
1. Human-Computer Interaction (3rd Edition) – Alan Dix, Janet Finlay, Gregory D. Abowd and Russell Beale; Prentice Hall (2003)
2. Interaction Design: Beyond Human Computer Interaction (5th Edition) - Yvonne Rogers, Helen Sharp, Jenny Preece; John Wiley & Sons (2019)
3. Designing the User Interface (5th Edition) - Schneiderman, B. and Plaisant, C.; Addison-Wesley (1998)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1							3					
CO 2										3		
CO 3							3					

Course Title: Human Computer Interaction Lab	**Credits:** 1.5
Course Code: 0688 4196/4296	**Contact Hours:** 3 hours/week

Rationale:
Motivates to enrich socio-technological discipline to bring the power of computers and communications systems to people in ways and forms that are both accessible interactively and useful in our working, learning, communicating, and recreational lives.

Objectives:
- To specify fundamental human and computational abilities and constraints and design requirements accordingly by using suitable HCI techniques.
- To design and evaluate the quality of a user interface using various methods to design a usable and useful interface.

Course Contents:
Introduction to Human-Computer Interaction, Task-centered system design, Design principles, graphical design by grids, design concepts, Prototype, IDM, Semiotic in HCI.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1		3										
CO 2					3							
CO 3				3								
CO 4			3									

CO1	Discover and apply theoretical knowledge and develop projects considering the criteria of HCI and meeting the principle.
CO2	Estimate and evaluate how developed systems (UI and Action) are violating different Heuristics of HCI.
CO3	Select and analyze the fundamental aspects of designing and evaluating interfaces.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook:
1. Human-Computer Interaction (3rd Edition) – Alan Dix, Janet Finlay, Gregory D. Abowd and Russell Beale; Prentice Hall (2003)

Course Title: Bioinformatics	**Credits:** 3.0
Course Code: 0688 4197/4297	**Contact hours:** 3 hours/week

Rationale:
In this course, students will learn fundamental concepts and methods in bioinformatics. This course will provide a certain level of understanding of molecular and cellular biology, and a working knowledge of bioinformatics algorithms and applications covering the topics of sequence similarity and alignments, evolutionary processes, protein structure, genome characteristics, and gene expression.

Objectives:
- To introduce students to the fundamentals of molecular and cellular biology, genetics, gene regulation, mutation, sequencing, and evolution.
- To help students develop skills that will enable them to formulate precise computational problems from biological problems.

- To help students develop skills that will enable them to design algorithms for biological problems
- To show students how to apply many of the primary predictive methods that are common in modern bioinformatics.
- To make the students have a practical and hands-on experience with modern bioinformatics tools and databases.
- To train the students in the basic theory and application of programs used for database searching, protein and DNA sequence analysis, prediction of protein function, and building phylogenetic trees.

Course Contents:
Cell concept: Cell structure, Cell cycle, Cell division, Chromosome**.** **Nucleic acids:** Structure and properties of different forms of DNA and RNA; DNA replication, Chargaff's Rule of Base Pairing. **Proteins:** Structure and classification, Amino Acids, Central dogma of molecular biology. **Genetic code:** A brief account. **Genetics:** Mendel's laws of inheritance, Organization of the genetic material of prokaryotes and eukaryotes, repetitive DNA, chromosome organization and banding patterns, the structure of gene - intron, exon, and their relationships, overlapping gene, regulatory sequence, Molecular mechanism of general recombination, gene conversion, Evolution and types of mutation, molecular mechanisms of mutation. **Introduction to Bioinformatics:** Definition and History of Bioinformatics, Bioinformatics Tools and Databases, Applications of Bioinformatics. **DNA Replication Origin:** Hidden Messages in the Replication Origin, Asymmetry of Replication, Deamination, The skew diagram, Finding DnaA Boxes in E. coli. **Motif Finding:** Scoring Motifs, Median String, Greedy Motif Search, Randomized Motif Search, Gibbs Sampling. **Genome Assembly:** String Reconstruction Problem, Overlap Graph, Hamiltonian path, Eulerian path and cycle, Euler's Theorem, de Bruijn Graph, Assembling Genomes from Read-Pairs. **Cyclopeptide Sequencing:** Discovery of Antibiotics, Non-ribosomal peptides, Mass spectrometry, Branch-and-Bound Algorithm for Cyclopeptide Sequencing, Theoretical to real spectra, The Spectral Convolution. **Sequence Alignment:** Non-Ribosomal Code, Manhattan Tourist Problem, Dynamic Programming, Scoring matrices, Global and Local Alignment, Space-Efficient Sequence Alignment, Multiple Sequence Alignment, The Blast family of programs. **Genome Rearrangement:** Random Breakage Model of Chromosome Evolution, Sorting by Reversals, Breakpoints, Breakpoint Graphs, 2-Break Distance Theorem, Rearrangement Hotspots, Fragile Breakage Model. **Phylogeny:** Distance Matrices into Evolutionary Trees, Additive Phylogeny, Ultrametric Evolutionary Trees, The Neighbor-Joining Algorithm, Character-Based phylogeny, The Small Parsimony and Large Parsimony Problem. **Gene Expression Analysis:** Genes Responsible for the Diauxic Shift, Clustering yeast genes, Clustering Principle, Farthest First Traversal, k-Means Clustering, The Lloyd Algorithm, Soft k-Means Clustering, Hierarchical Clustering. **Single Nucleotide Polymorphisms:** Ohdo Syndrome, Multiple Pattern Matching, Trie Construction, Suffix Trees, Suffix Arrays, The Burrows-Wheeler Transform, Pattern Matching with the Burrows-Wheeler Transform, Speeding Up Burrows-Wheeler Pattern Matching. **Hidden Markov Models in Bioinformatics:** Classifying the HIV Phenotype, Limitations of sequence alignment, Finding CG-Islands, Hidden Markov Models, The Viterbi algorithm, Profile HMMs for Sequence Alignment, Classifying proteins with profile HMMs, Learning the Parameters of an HMM, Soft Decisions in Parameter Estimation, Baum-Welch Learning. **Paleontology:** Computational Paleontology, Peptide Sequencing, Decoding an Ideal and Real Spectrum, Peptide Identification and the Infinite Monkey Theorem, Spectral Dictionaries, T. rex Peptides, The dinosaur DNA controversy.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Apply reasoning about core biological concepts related to bioinformatics with emphases on molecular and cellular biology, genetics, gene regulation, mutation, sequencing, and evolution.
CO 2	Formulate rigorous computational problems from complex biological problems.
CO 3	Design, implement and evaluate computational algorithms in relation to the contexts of molecular and cellular biology and genomics research.
CO 4	Analyze and evaluate bioinformatics data to discover patterns, critically evaluate conclusions, and generate predictions for subsequent experiments.
CO 5	Exploit modern bioinformatics tools and databases for analyzing the biological sequences to infer functional and evolutionary relationships between sequences as well as help identify its category.

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook
1. Bioinformatics Algorithms: An Active Learning Approach by Phillip Compeau and Pavel Pevzner.
2. Bioinformatics with Python Cookbook by Tiago Antao
3. Understanding Bioinformatics by Market Zvelebil, Jeremy O. Baum

4. Bioinformatics for Biologists by Pavel Pevzner and Ron Shamir.

Course Title: Bioinformatics Lab	**Credits:** 1.5
Course Code: 0688 4198/4298	**Contact hours:** 3 hours/week

Rationale:

In this course, students will learn how to implement and apply algorithms from the bioinformatics course in the laboratory. This course will provide practical and hands-on experience in designing, evaluating, and implementing bioinformatics algorithms as well as modern bioinformatics tools and databases.

Objectives:

- To help students develop skills that will enable them to design and implement algorithms for biological problems
- To show students how to apply many of the primary predictive methods that are common in modern bioinformatics.
- To make the students have a practical and hands-on experience with common bioinformatics tools and databases.
- To train the students in the basic theory and application of programs used for database searching, protein and DNA sequence analysis, prediction of protein function, and building phylogenetic trees.

Course Contents:

Python Basics: Variables, Expressions, Conditional & Iteration Statements, Functions, String, List, Dictionary, Tuples, Numpy, and Pandas. **Biopython:** Seq and SeqRecord objects, Complement, Reverse Complement, Transcribe, Translate, Codon Table, GC Content, Molecular Weight, Entrez & ExPASy Database Search & Access, Parsing Fasta, Genbank Files. **DNA Replication Origin:** Finding pattern, all occurrences of pattern, most frequent word, most frequent word with mismatches, patterns forming a clump, position in a genome minimizing skew, Drawing the skew diagram, Computing Hamming Distance, Implementing PatternToNumber and NumberToPattern. **Motif Finding:** Find a Profile-most Probable k-mer in a String, Find a Median String, Implement DistanceBetweenPatternAndStrings, MotifEnumeration, GreedyMotifSearch, GreedyMotifSearch with Pseudocounts, RandomizedMotifSearch, GibbsSampler algorithms. **Genome Assembly:** Generate the k-mer Composition of a String, Reconstruct a String from its Genome Path, Construct the Overlap Graph, the De Bruijn Graph, Find an

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3			3									
CO 4				3								
CO 5					3							

Eulerian Cycle & Path in a Graph, Reconstruct a String from its k-mer and Paired Composition, Find a k-Universal Circular String, Generate Contigs from a Collection of Reads, Construct a String Spelled by a Gapped Genome Path, Generate All Maximal Non-Branching Paths in a Graph. **Cyclopeptide Sequencing:** Translate an RNA String into an Amino Acid String, Find Substrings of a Genome Encoding a Given Amino Acid String, Generate the Theoretical Spectrum of a Cyclic & Linear Peptide, Compute the Number of Peptides for a Given Total Mass, Find a Cyclic Peptide with Theoretical Spectrum Matching an Ideal Spectrum, Compute the Score of a Cyclic & Linear Peptide, Generate the Convolution of a Spectrum, Implement Leaderboard Cyclopeptide Sequencing, Convolution Cyclopeptide Sequencing algorithms. **Sequence Alignment:** Find the Length of a Longest Path in a Manhattan-like Grid & DAG, the Longest Common Subsequence, the Edit Distance, the Global & Local Alignment of Two Strings, Align Two Strings Using Affine Gap Penalties, Find a Middle Edge in an Alignment Graph in Linear Space, Align Two Strings Using Linear Space, Find a Highest-Scoring Multiple Sequence Alignment. **BLAST:** Search, Access, Parse & Interpret Results of the NCBI BLAST tool via Web portal and with the python program. **Genome Rearrangement:** Compute the Number of Breakpoints in a Permutation, the 2-Break Distance Between a Pair of Genomes, Find a Shortest Transformation of One Genome into Another by 2-Breaks, Find All Shared k-mers of a Pair of Strings, Implement Greedy Sorting, ChromosomeToCycle, CycleToChromosome, ColoredEdges, GraphToGenome, 2-BreakOnGenomeGraph, 2-BreakOnGenome algorithms. **Phylogeny:** Compute Distances Between Leaves, Compute Limb Lengths in a Tree, Implement AdditivePhylogeny, UPGMA, the Neighbor-Joining, and SmallParsimony algorithms. **Gene Expression Analysis:** Implement FarthestFirstTraversal, Compute the Squared Error Distortion, and Implement the Lloyd Algorithm for k-Means Clustering, Soft k-Means Clustering, and Hierarchical Clustering. **Single Nucleotide Polymorphisms:** Construct a Trie from a Collection of Patterns, the Suffix Tree, Array, Burrows-Wheeler Transform, Partial Suffix Array of a String, Find the Longest Repeat in a String, the Longest Substring Shared & Shortest Non-Shared by Two Strings, Pattern Matching with the Suffix Array, Generate the Last-to-First Mapping of a String, Implement TrieMatching, BWMatching, Implement BetterBWMatching, TreeColoring algorithms. **Hidden Markov Models in Bioinformatics:** Compute the Probability of a Hidden Path, an Outcome Given a Hidden Path, a String Emitted by an HMM, Implement the Viterbi, Viterbi Learning & Baum-Welch Learning Algorithm, Construct a Profile HMM & Profile HMM with Pseudocounts, Perform a Multiple Sequence Alignment with a Profile HMM, Estimate the Parameters of an HMM, Solve the Soft Decoding Problem. **Paleontology:** Construct the Graph of a Spectrum, Implement DecodingIdealSpectrum, Convert a Peptide into a Peptide Vector, Convert a Peptide Vector into a Peptide, Sequence a Peptide, Find a Highest-Scoring Peptide in a Proteome against a Spectrum, Implement PSMSearch, Compute the Size of a Spectral Dictionary, Compute the Probability of a Spectral Dictionary, Find a Highest-Scoring Modified Peptide against a Spectrum.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Employ python and its modules to solve bioinformatics problems.
CO 2	Design, implement and evaluate computational algorithms in relation to the contexts of molecular and cellular biology and genomics research.
CO 3	Analyze and evaluate bioinformatics data using python and its modules to discover patterns, critically evaluate conclusions, and generate predictions for subsequent experiments.
CO 4	Exploit modern bioinformatics tools and databases for analyzing the biological sequences to infer functional and evolutionary relationships between sequences as well as help identify its category.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3									
CO 3				3								
CO 4					3							

Textbook

1. Bioinformatics Algorithms: An Active Learning Approach by Phillip Compeau and Pavel Pevzner.
2. Bioinformatics with Python Cookbook by Tiago Antao
3. Understanding Bioinformatics by Market Zvelebil, Jeremy O. Baum
4. Bioinformatics for Biologists by Pavel Pevzner and Ron Shamir.

Course Title: Mobile and Wireless Communication	**Credits:** 3.0
Course Code: 0714 4171/4271	**Contact Hours:** 3 hours/week

Rationale:

The purpose of this course is to provide students knowledge about communication and data transfer between devices without the need for physical, wired connections.

Objectives:

- To develop a profound knowledge of wireless and mobile communication.
- To analyze hands-on problems in wireless communication.
- To design wireless and cellular networks.

Course Contents:

Aspects of radio wave propagation for fixed and mobile communication systems, and cellular system design. Large-scale and small-scale propagation models, multipath fading, link-budget, interference and frequency reuse, multiple access schemes and system capacity. Trunking and grade of service, wireless network planning and operation. Architecture and operation of 2G cellular mobile systems, 2.5 G and 3G technologies. Special techniques/Diversity, Equalization, Interleaving, and Smart Antenna.

Course Learning Outcomes, CO

After the successful completion of the course, students will be able to:

CO 1: demonstrate knowledge on : cellular concepts like frequency reuse, fading, equalization, GSM, CDMA

CO 2: apply the concept of GSM in real time applications..

CO 3: compare different multiple access techniques in mobile communication

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3		2								
CO 3			3				1					

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3		2								
CO 3			3				1					

Textbook
1. Mobile and Wireless Communications: An Introduction- Gordon A. Gow, Richard K. Smith.
2. Communications Engineering: Essentials for Computer Scientists and Electrical Engineers- Richard Chia Tung Lee, Mao-Ching Chiu, Jung-Shan Lin

Course Title: Mobile and Wireless Communication Lab	**Credits:** 1.5
Course Code: 0714 4172/4272	**Contact Hours:** 3 hours/week

Rationale:
The purpose of this course is to provide students knowledge about communication and data transfer between devices without the need for physical, wired connections.

Objectives:
- To develop a profound knowledge of wireless and mobile communication.
- To analyze hands-on problems in wireless communication.
- To design wireless and cellular networks.

Course Contents:
Aspects of radio wave propagation for fixed and mobile communication systems, and cellular system design. Large-scale and small-scale propagation models, multipath fading, link-budget, interference and frequency reuse, multiple access schemes and system capacity. Trunking and grade of service, wireless network planning and operation. Architecture and operation of 2G cellular mobile systems, 2.5 G and 3G technologies. Special techniques/Diversity, Equalization, Interleaving, and Smart Antenna.

Course Learning Outcomes, CO
After the successful completion of the course, students will be able to:

CO 4: demonstrate knowledge on : cellular concepts like frequency reuse, fading, equalization, GSM, CDMA

CO 5: apply the concept of GSM in real time applications..

CO 6: compare different multiple access techniques in mobile communication

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook
3. Mobile and Wireless Communications: An Introduction- Gordon A. Gow, Richard K. Smith.
4. Communications Engineering: Essentials for Computer Scientists and Electrical Engineers- Richard Chia Tung Lee, Mao-Ching Chiu, Jung-Shan Lin

Course Title: Internet of Things	**Credits:** 3.0
Course Code: 0714 4173/4273	**Contact Hours:** 3 hours/week

Rationale:
This course is designed to provide a general introduction to Internet of Things. It will help the students to develop skills and experiences in IoT concepts such as sensing, actuation and communication. After completion of the course students will be able to build IoT applications and devices.

Objectives:
- To give students an insight about the Internet's revolution in communication with wireless sensors, software, and actuators.
- To provide the students with knowledge of the fundamental components of the Internet of Things (IoT).
- To introduce students to the technology that is used to create IoT devices. What kind of distributed systems are required to support them, how they communicate, and how they store data.
- To assist students in gaining knowledge and experience necessary for building IoT prototypes.

Course Contents:
Introduction to IoT: Fundamentals, Architecture and Protocols.Overview of IoT components and Technologies. Sensing, Actuation, Networking, Sensor Networks. **M2M:**Machine-to-Machine Communications, Difference between IOT and M2M, Software defined Networking, Network Function Virtualization, Need for IOT Systems Management. **Arduino Simulation Environment**: Introduction to Arduino Programming, Arduino Uno Architecture, Setup the IDE, Writing Arduino Software, Integration of

Sensors and Actuators with Arduino.**Developing IoTs**: Introduction to Python programming, Introduction to Raspberry Pi, Data Handling and Analytics.**Cloud Computing for IoTs:**Basics of Cloud Computing, Cloud services - SaaS, PaaS, IaaS; IoT Cloud platforms, Fog Computing.**Domain specific applications of IoT:**Connected Vehicles, Smart Grid, Industrial IoT.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Illustrate the concepts of Internet of Things.
CO 2	Apply basic protocols in wireless sensor network.
CO 3	Design IoT applications in different domain and be able to evaluate their effectiveness.
CO 4	Develop IoT Devices.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2	3		3									
CO 3			3							3	3	
CO 4					3			3	3		3	3

Textbook

1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases by Pethuru Raj and Anupama C. Raman (CRC Press).
2. Internet of Things: A Hands-on Approach by ArshdeepBahga and Vijay Madisetti (Universities Press)

Course Title: Internet of Things Lab	**Credits:** 1.5
Course Code: 0714 4174/4274	**Contact Hours:** 3 hours/week

Rationale:

This course is designed to provide a general introduction to Internet of Things. It will help the students to develop skills and experiences in IoT concepts such as sensing, actuation and communication. After completion of the course students will be able to build IoT applications and devices.

Objectives:

- To provide the students with knowledge of thefundamental components of the Internet of Things (IoT).
- To introduce students to the technology that is used to create IoT devices. What kind of distributed systems are required to support them, how they communicate, and how they store data.
- To assist students in gaining knowledge and experience necessary for building IoT prototypes.

Course Contents:

Arduino Simulation Environment:Setup the IDE, Writing Arduino Software, Integration of Sensors, and Actuators with Arduino.**Developing IoTs**: Introduction to Python programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi, Data Handling and Analytics.**Domain specific developments of IoT Applications and Devices.**

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Integrate Sensors, and Actuators with Arduino.
CO 2	ImplementIoT applications with Python and Raspberry Pi.
CO 3	Design domain specific IoT applications and evaluate their effectiveness.
CO 4	Develop IOT Devices.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1			3		3							
CO 2			3		3				3			
CO 3			3		3		3		3		3	
CO 4								3	3		3	

Textbook

1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases by Pethuru Raj and Anupama C. Raman (CRC Press).

2. Internet of Things: A Hands-on Approach by ArshdeepBahga and Vijay Madisetti (Universities Press)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2			3		3							
CO 3			3		3							
CO 4				3								
CO 5			3		3							
CO 6						3	3					

Course Title: VLSI Design	**Credits:** 3.0
Course Code: 0714 4177/4277	**Contact Hours:** 3 hours/week

Rationale:
The course is designed to provide the fundamental concepts required for developing any VLSI systems in any organization.

Objectives:
- To learn basic CMOS Circuits.
- To learn CMOS process technology.
- To learn techniques of chip design using programmable devices.
- To learn the concepts of designing VLSI Subsystems.

Course Contents:
IC, Generations, Moor's Law. MOS, BJT, Design Process, MOS Structure, NMOS, PMOS, MOS Conduction, Threshold Voltage. Ids vs Vds characteristics for NMOS, Characteristics Equation for PMOS Devices. BJT: Common base, Common emitter. Principles of Inverters, NMOS Inverter with Resistor Load, NMOS Inverter with NMOS Depletion Transistor Load. NMOS pass transistor, Pass transistor, NMOS ratio less inverter, Buffer gate chain, NMOS non-inverting super buffer. NMOS inverting super buffer, MOS Process, NMOS Processing. CMOS pass Gate, I versus Vout curves for T1 and T2, Characteristic curves for CMOS inverter. Circuit diagram, stick diagram, Geometric Diagram of CMOS, NMOS, PMOS. MOS Logical Circuit Design.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Identify the various IC fabrication methods
CO2	Express the Layout of simple MOS circuit using Lambda based design rules
CO3	Apply the Lambda based design rules for subsystem design
CO4	Differentiate various FPGA architectures
CO5	Design an application using Verilog HDL
CO6	Concepts of modeling a digital system using Hardware Description Language

Mapping of Course Learning Outcomes to Program Learning Outcomes

Textbook
1. Electronic Devices and Circuits Theory by Robert L.Boylestad.
2. VLSI Technology by S.M.SZE.E. Woods, Steven L. Eddins

Course Title: VLSI Design Lab	**Credits:** 1.5
Course Code: 0714 4178/4278	**Contact Hours:** 3 hours/week

Rationale:
This course provides practical knowledge about the design of various digital circuits using different VLSI simulation software tools and develops understanding about approximation problem and ability to use approximation concept in real world problem.

Objectives:
- To learn basic CMOS Circuits.
- To learn CMOS process technology.
- To learn techniques of chip design using programmable devices.
- To learn the concepts of designing VLSI Subsystems.

Course Contents:
Design basic MOS circuits: Design and simulate NMOS and PMOS. **CMOS design**: Design and simulate CMOS technology. **Design of inverters**: Design and simulate CMOS inverter technology. **Logic gate implementation**: Design of logic gates using CMOS technology. **Design and simulate combinational sequential digital circuit**. **Design of 8 Bit Adders:** design of ripple carry adder using Verilog HDL, design carry save

adder using Verilog HDL, design carry select adder using Verilog HDL, BCD adder realization in Verilog HDL.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	design and simulate the NMOS, PMOS and CMOS circuits and compare performances using Microwind
CO2	design and simulate list of combinational sequential digital circuits using Modelsim and Xilinx –VHDL language
CO3	design and simulate the adder using Modelsim Xilinx-Verilog language

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1					3							
CO 2		3										
CO 3							3					

Textbook

1. Electronic Devices and Circuits Theory by Robert L.Boylestad.
2. VLSI Technology by S.M.SZE.E. Woods, Steven L. Eddins

Course Title: Contemporary Course on Computer Science and Engineering I	**Credits:** 3.0
Course Code: 0611/0612/0613/0619 4187	**Contact Hours:** 3 hours/week

This course covers a contemporary title in Computer science and Engineering decided by the department.

Course Title: Contemporary Course on Computer Science and Engineering I Lab	**Credits:** 3.0
Course Code: 0611/0612/0613/0619 4188	**Contact Hours:** 3 hours/week

Laboratory works based on 0611/0612/0613/0619 4187

Course Title: Contemporary Course on Computer Science and Engineering II	**Credits:** 3.0
Course Code: 0611/0612/0613/0619 4287	**Contact Hours:** 3 hours/week

This course covers a contemporary title in Computer science and Engineering decided by the department.

Course Title: Contemporary Course on Computer Science and Engineering II Lab	**Credits:** 3.0
Course Code: 0611/0612/0613/0619 4288	**Contact Hours:** 3 hours/week

Laboratory works based on 0611/0612/0613/0619 4287

Detailed Syllabus
Non-Major Courses
(offered by other departments)

First Year First Semester

Course Title: Effective Communication in English	**Credits:** 2.0
Course Code: 0231 1101D	**Contact Hours:** 2 hours/week

Rationale:
This course is expected to develop two basic skills i.e. reading and writing. A variety of reading strategies and texts will be used to effectively develop first year students' academic reading skills thereby facilitating their future study. Also, the course focuses on developing the writing skills of students by familiarizing them with grammar rules, providing them with practice and enabling them to demonstrate the accurate use of grammar in their writing.

Course Objectives
(i) To enable students to write with accuracy;
(ii) To facilitate effective and comprehensible writing;
(iii) To raise awareness of common errors that occur in writing;
(iv) To develop students' ability to understand write-ups on issues of general concern;
(v) To improve the vocabulary of learners for effective communication.

Course Contents
Reading: Different Reading Strategies, Guessing Meaning from the Context, Critical Reading (Analyze), Critical Reading (Synthesize), Critical Reading (Evaluate), Annotation, Summary Writing. **Writing:** Forms and functions of different word categories (noun, verb, adjective, etc.), Aspects and uses of tense, Subject-verb agreement, Use of infinitive, gerund, present participle, past participle, modals, causatives, conditionals, subjunctives, modals, Use of sentence connectors/ cohesion markers/ punctuation, gender fair language, euphemistic expression, formal and informal language, collocation, idiomatic expression, error correction, removing wordiness, Effective combination of sentences (simple, complex, compound), Developing a paragraph

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	apply grammar rules
CO 2	express oneself correctly by using appropriate words, phrases, sentences or ideas
CO 3	critically reflect on a text (grasp abstract ideas and interpret them effectively, arrive at well-reasoned conclusions and solutions)
CO 4	Create using earned knowledge both independently and in collaboration with peer groups

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1						3						
CO 2							3					
CO 3										3		
CO 4												3

Materials
- A selection of 08-10 editorials and reports from newspapers/magazines/journals,etc.
- Reading texts in New Headway Upper Intermediate Student's Book (Current edition)
- Selected passages from recommended books
- A selection of other materials may be supplied as handouts by the instructor as necessary

Textbook
1. Tibbits, E. E., editor. Exercises in Reading Comprehension. Longman, 2013.
2. Liz and John Soars. New Headway Upper Intermediate Student's Book. Oxford University Press, 2014.
3. Payle, Michael. Cliff's TOEFL Preparation Guide. 12th ed., Cliffs Notes Inc., 2019.
4. Other resources recommended by course instructors

Course Title: English Language Lab - I	**Credits:** 1.0
Course Code: 0231 1102D	**Contact hours:** 2 hours/week

Course Rationale

This course is designed to improve the speaking and listening skills of students in the English language. Emphasis is laid on proper pronunciation for accurate articulation and recognition of speech sounds as well as correct stress, intonation and language use in varied situations.

Course Objectives

(i) To enable students' understanding of the variations in pronunciation;
(ii) To teach proper pronunciation and accurate articulation;
(iii) To facilitate appropriate stress and intonation in speech;
(iv) To encourage use of English effectively in everyday situations;
(v) To ensure overall improvement of oral communication through listening and speaking.

Course Content

Speaking: Articulators, English Phonetic Alphabet (British and American) and International Phonetic Alphabet (IPA), Stress rules of English, Intonation rules and functions of intonation, Communication styles and cultural context, Fluency, mistakes, misunderstandings, audience, taboos, self-esteem, confidence, Activities: dialogue, debate, extempore speech, interview, role-play.

Listening: Basics of listening, Various types of pronunciation, IPA, RP, transcription, Different accents and intonation patterns, Activities for meaning-focused listening, Information transfer strategies, Listening practice through selection of audio clips

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	read the symbols of the International Phonetic Alphabet used to represent the sounds of the English language
CO 2	apply appropriate intonation and stress patterns in English words and sentences
CO 3	interpret information accurately
CO 4	collaborate and apply intonation and stress patterns.
CO 5	produce continuous speech clearly and convincingly

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1						3						
CO 2							3					
CO 3									3			
CO 4										3		
CO 5												3

Textbook

1. Anderson, Anne C., etal.Listening.Oxford University Press, 1988.
2. Anderson, Kenneth, et al. Study Speaking. Cambridge University Press, 2007.
3. Hancock, Mark. English Pronunciation in Use. Cambridge University Press, 2004.
4. Jones, Daniel. Cambridge English Pronunciation Dictionary. Cambridge University Press, 2011. Richards, Jack C., and David Bohlke. Speak Now: 1. Oxford University Press, 2013.
5. Richards, Jack C., et al. Person to Person. Oxford University Press, 2007.
6. Roach, Peter. English Phonetics and Phonology. Cambridge University Press, 2009.

Course Title: Coordinate Geometry And Linear Algebra		**Credits:** 3.0
Course Code: 0541 1101D	**Contact Hours:** 3 hours/week	

Rationale:

In an increasingly complex world, mathematical thinking, understanding, and skill are more important than ever. This course will show students how to simplify many types of complex problems using matrix algebra and vector geometry. Students who major in the sciences or engineering are often required to study linear algebra. This course provides a solid foundation for further study in mathematics, the sciences, and engineering.

Course Objectives:

- Engage students in sound mathematical thinking and reasoning
- Provide a setting that prepares students to read and learn mathematics on their own
- Enhance and reinforce the student’s understanding of concepts through the use of technology when appropriate

Course Contents:

Coordinate geometry: Coordinate geometry of two dimensions: Change of axes; transformation of co-ordinates; pair of straight lines; general equation of second degree. Coordinate geometry of three dimensions: System of co-ordinates; distances of two points; section formula; projection; direction cosines; equations of planes and straight lines. **Matrix:** Matrix and matrix operations; different types of matrices; algebraic operations on matrices; cofactors and minors; determinant of a square matrix; adjoint and inverse of a matrix; elementary transformation of matrices; normal and canonical form of a matrix; rank of a matrix; the row-reduced form of a matrix and rank; equivalent systems of linear equations; the general solution of a system of linear equations; homogeneous systems; eigenvalues and eigenvectors; diagonalization of matrices. **Vector space:** Vector spaces and subspaces; linear dependence and independence; spanning set and basis; coordinates and dimension; null space, row space and column space; change of basis. **Linear transformations:** Linear transformations; composition of transformations; matrix representation; change of basis; diagonalization representation of a linear transformation by a diagonal matrix; the eigenvalues and eigenvectors of a symmetric matrix; quadratic form; functions of a square matrix. Inner product spaces: Definition and examples; Cauchy-Schwartz inequality; orthogonality; orthonormal basis and Gram-Schmidt process.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	Interpret quantitative data verbally, graphically, symbolically and numerically
CO 2	Determine the size, transpose, inverse, rank, and LU-factorization of a matrix
CO 3	Perform matrix operations and solve matrix equations
CO 4	Perform vector operations for vectors in R, compute vector additions and scalar multiplications
CO 5	Use mathematical concepts in problem-solving through integration of new material and modeling

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			2								
CO 2				3								
CO 3		2										
CO 4				3								
CO 5										3		

Recommended Books

1. Howard Anton and Chris Rorres: Elementary linear algebra applications
2. Steven J. Leon: Linear algebra with applications, Prentice Hall, 1998
3. Rahman and Bhattacharjee: Co-ordinate geometry of two and three dimensions
4. Loney, S. L.: Coordinate Geometry of Two dimensions

Course Title: Electrical Circuits	**Credits:** 3.0
Course Code: 0713 1109D	**Contact hours:** 3 hours/week

Rationale:

This course is an introductory course that thoroughly explains different circuit analysis techniques to analyze simple and complex circuits. It also provides ideas about AC networks, including the use of phasor and impedance diagrams. This course endeavors to build on this knowledge and further expand students' skills in analyzing and designing circuits involving transistors, diodes, operational amplifiers and basic logic gates. Finally, the course focuses on developing fundamental ideas and concepts on electrical equipment and electronic devices.

Course Objectives:

- To disseminate knowledge about electrical charge, voltage, current and power.
- To give the idea of basic concepts of DC circuit behavior.
- Help the students to conceptualize the fundamental theorems of circuit analysis.
- To help the students develop skills to solve mathematical problems of simple and complex electrical circuits.
- To familiarize the students with the basics of AC networks.
- To teach the modelling and analysis of single-phase RLC circuits for impedances, voltages, currents, powers, and phase shift.

Course Contents:

Circuit variables and elements: Voltage, current, power, energy, independent and dependent sources, and resistance. **Basic laws:** Ohm's law, Kirchhoff's current and voltage laws. **Simple resistive circuits**: Series and parallel circuits, voltage and current division, wye-delta transformation. **Techniques of circuit analysis:** Nodal and mesh analysis including super node and super mesh. **Network theorems:** Source transformation, Thevenin's, Norton's and superposition theorems with applications in circuits having independent and dependent sources, maximum power transfer condition and reciprocity theorem. **Energy storage elements:** Inductors and capacitors, series parallel combination of inductors and capacitors. **Responses of RL and RC circuits:** Natural and step responses. **Sinusoidal functions:** Instantaneous current, voltage, power, effective current and voltage, average power, phasors and complex quantities, impedance, real and reactive power, power factor. **Analysis of single-phase AC circuits:** Series and parallel RL, RC and RLC circuits, nodal and mesh analysis, application of network theorems in AC circuits.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	Understand fundamental terms related to electrical circuits.
CO 2	Summarize the basics of energy storage elements, AC circuit concepts and responses.
CO 3	Interpret the basic circuit laws and circuit analysis techniques.
CO 4	Apply circuit analysis techniques to analyze DC and AC circuits.
CO 5	Design and analyze DC as well as single-phase AC circuits.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2	3											
CO 3	3											
CO 4	3	2			2							
CO 5	3	2			2							

Recommended Books

1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku
2. Introductory Circuit Analysis by Robert L.Boylestad.

Course Title: Electrical Circuits Lab	**Credits:** 1.5
Course Code: 0713 1110D	**Contact hours:** 3 hours/week

Rationale

In this course, students will perform experiments to verify practically the theories and concepts learned in 0713 1109D. Theoretical knowledge is complete with hands-on experiments using the essential components and measuring devices used in electrical circuit analysis. This course teaches the fundamentals of electrical circuits, the application of circuit laws, theorems and measuring techniques for DC circuits. It contains a broad idea of single-phase power systems with various resistive and reactive loads. It also familiarizes the relationship between real, apparent and reactive power - including phasor and impedance diagrams, methods of measuring power, and calculation of power factor.

Course Objectives

The objectives of the course are to-

- Provide students the capability of implementing different real-life dc circuits.
- Enable students with the network analysis techniques to solve different types of circuits.
- Teach the voltage, current and load relationship in a network.
- Facilitate necessary knowledge about transient analysis and steady state analysis of a capacitor and inductor network.
- Help students to develop the ability in building AC electrical circuits and perform experiments on them..

Course Contents:
In this course students will perform experiments to verify practically the theories and concepts learned in EEE-109.
Lab 1-2: To familiarize students with the operation of different electrical instruments.
Lab 3-8: To verify the following theorems: KCL and KVL theorem, Superposition theorem, Thevenin's theorem, Norton's theorem and Maximum power transfer theorem.
Lab 9-10: RL and RC response.
Lab 11: Study the frequency response of an RLC circuit and find its resonance frequency.
Lab 12-13: Connecting basic electrical elements such as fan, bulb, calling bell etc. from the 220 V AC single phase supply.
Lab 14: Relevant application based on EEE 0713-1109D.

Course Learning Outcomes:
After the successful completion of the course, the student will be able to-

CO 1	Differentiate the operating principle of various electrical instruments and measuring devices.
CO 2	Design experiments to interpret different types of circuit analysis theorem and laws.
CO 3	Explain the idea about complex circuit networks.
CO 4	Interpret transient responses of the capacitor and inductor circuits
CO5	Demonstrate team-based communication skills and apply these in practical life

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			2								
CO 2				3								
CO 3		2										
CO 4				3								
CO 5										3		

Recommended Books
1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku
2. Introductory Circuit Analysis by Robert L.Boylestad

First Year Second Semester

Course Title: Mechanics, Waves, Heat and Thermodynamics		**Credits:** 3.0
Course Code: 0533 1203D	**Contact Hours:** 3 hours/week	

Rationale of the Course
This course provides basic concepts on Motion, Wave frequencies, Sound signals through Fourier Theorem, Heat and Thermodynamics. This course helps students to apply this knowledge to computer programming, related to physical world.

Course Objectives
- to facilitate necessary knowledge about Newtonian mechanics and problem-solving technique.
- to accumulate basic ideas on waves behavior and Fourier theorem.
- to make the students understand the basic principles of thermodynamics.
- to get knowledge on the radiation.

Course Content
Mechanics: motion in two dimensions, projectile motion, Newton's laws of motion, central forces and gravitation, Kepler's law. **Waves:** simple harmonic motion, damped and forced harmonic vibrations, waves in elastic media. Fourier's theorem and application. **Heat, Thermodynamics and Radiation:** principles of thermometry, zeroth law of thermodynamics. kinetic theory of gases, first and second law of thermodynamics, entropy, blackbody radiation. Wein's law, Planck's law.

Course Learning Outcomes
After the successful completion of the course, students will be able to:

CO1	explain motion, cause of motion, projectile motion and applying the idea in different cases.
CO2	apply Kepler’s laws to predict path of the Planets and calculate revolving time around the Sun.
CO3	perform the details calculation about simple harmonic motion for free, damped and forced vibrations.
CO4	apply laws of thermodynamics and explain the black-body radiation.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3		2		3	2					3
CO2	3	3		2		3	2					3
CO3	3	3		2		3	2					3
CO4	3	3		2		3	2					3

Recommended Books

1. Halliday, D. and Resnick, R.: Physics (Vol. I and Vol II)
2. Puri, S. P.: Fundamentals of Vibrations and Waves
3. Saha and Srivastava: A Treatise of Heat

Course Title: Calculus	**Credits:** 3.0
Course Code: 0541 1203D	**Contact hours:** 3 hours/week

Rationale

The focus and themes of the Calculus course address the most important foundations for applications of mathematics in science, engineering and commerce. The course emphasizes the key ideas and historical motivation for calculus, while at the same time striking a balance between theory and application, leading to a mastery of key threshold concepts in foundational mathematics.

Course Objectives

- Establish the fundamental theorems and applications of the calculus of single variable functions
- Help them explore the concepts, properties and aspects of the differential and integral calculus of single variable functions
- Provide students with the mathematical tools necessary for more advanced fields

Course Contents:

Differential Calculus: Functions of a real variables and their plots; limit; continuity and derivatives; physical meaning of derivative of a function; Leibnitz Theorem; Rolle's Theorem; mean value theorem and Taylor’s theorem (statement only).Taylor’s and Maclaurin’s series and expansion of functions; maximum and minimum values of functions; functions of two or three variables; partial and total derivatives. **Integral Calculus**: Physical meaning of integration of a function; integration as an inverse process of differentiation; different techniques of integrations; definite integrate as the limit of a sum and as an area; definition of Riemann integrals; fundamental theorem of integral calculus and its application to definite integrals; reduction formula; improper integrals; double integration; evaluation of area and volume by integration. **Differential Equations**: Definition and solution of ordinary differential equation; first order ordinary differential equation; second order ordinary linear differential equation with constant coefficients; initial value problems.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	Demonstrate knowledge of basic pre-calculus concepts and skills
CO 2	Evaluate limits and Recognize continuity and use the properties of continuous functions
CO 3	Find derivatives of algebraic and trigonometric functions using the definition or basic rules of differentiation
CO 4	Develop and practice methods of differential calculus with applications
CO 5	Develop and practice methods of the integral calculus

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			2								
CO 2				3								
CO 3		2										
CO 4				3								
CO 5										3		

Recommended Books

1. Das and Mukherjee: Differential Calculus
2. Das and Mukherjee: Integral Calculus
3. M.R. Spiegel: Advanced Calculus
4. J. Edwards: Differential Calculus
5. J. Edwards: Integral Calculus
6. R.A. Sardar: Differential Calculus
7. S. L. Ross: Differential equations

Course Title: Electronic Devices & Circuits	**Credits:** 3.0
Course Code: 0714 1211D	**Contact hours:** 3 hours/week

Rationale

This course endeavors to build on this knowledge and further expand students' skills in analyzing and designing analogue circuits involving transistors and diodes. The course covers the basic principles of operation and device characteristics of diodes, Bipolar Junction Transistors (BJT), Junction Field Effect Transistors (JFET) and Metal Oxide Semiconductor Field Effect Transistors (MOSFET) that underpin the analysis, design, and implementation of analogue circuits. Multi-stage amplifiers using BJT, and FETs further enhanced the course. Upon completion, students should be able to construct, analyze, verify, and troubleshoot analog circuits using appropriate techniques and test equipment..

Course Objectives:

- Acquaint the students with basic operations, device, and circuit characteristics of diodes, BJT, FET and Op-Amp.
- To enhance skill and knowledge in analyzing and designing analogue circuits such as amplifiers.
- To facilitate necessary knowledge about DC and AC analysis of different amplifier circuits.
- To help students conceptualize semiconductor theory.

Course Contents:

P-N junction diode as a circuit element: Intrinsic and extrinsic semiconductors, operational principle of p-n junction diode, contact potential, current-voltage characteristics of a diode, diode circuits: Half wave and full wave rectifiers, rectifiers with filter capacitor, characteristics of a Zener diode, clamping and clipping circuits. **Bipolar Junction Transistor (BJT) as a circuit element:** Current components, BJT characteristics and regions of operation, BJT as an amplifier, biasing the BJT for discrete circuits, small signal equivalent circuit models, BJT as a switch. **Field Effect Transistor (FET) as circuit element:** Structure and physical operation of a JFET, depletion type MOSFET and an enhancement type MOSFET, threshold voltage, Body effect, current-voltage characteristics of an enhancement MOSFET, and biasing discrete and integrated MOS amplifier circuits, single-stage MOS amplifiers, MOSFET as a switch, CMOS inverter.**Operational amplifiers (Op-Amp):** Properties of ideal Op-Amps, non-inverting and inverting amplifiers, inverting integrators, differentiator, weighted summer, and other applications of Op-Amp circuits.

.Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	Explain the basic operation of diode, BJT, FET, and Op-Amp.
CO 2	Design different amplifier circuits utilizing BJT, FET and Op-Amps.
CO 3	Perform DC biasing, and small signal analysis of the amplifier circuits.
CO 4	Differentiate between the switching and amplifying actions of BJT and FET.

CO 5	Interpret the concept on working principle and application of the photodiodes.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2		3										
CO 3	2		3		2							
CO 4	3											
CO 5	2	2					2					1

Recommended Books

1. Electronic Devices and Circuit Theory by Robert L. Boylestad and Louis Nashlesky
2. Microelectronic Circuits- Sedra/Smith
3. Digital logic and Computer Design – M. Morris Mano

Course Title: Electronic Devices & Circuits Lab	**Credits:** 1.5
Course Code: 0714 1212D	**Contact hours:** 3 hours/week

Rationale

In this course, students will perform experiments to verify practically the theories and concepts learned in 0714 1211D. Theoretical knowledge is complete with hands-on experiments using the essential components and measuring devices. This introductory experimental laboratory explores the design, construction, and debugging of analog electronic circuits. The experiments have been designed to investigate the performance characteristics of diodes, transistors, JFETs, MOSFETS and Op-Amps. In addition, the course provides opportunities to simulate real-world problems (given as an assignment) and solutions involving tradeoffs and engineering judgment.

Course objectives:

● Acquaint students with the basic idea about implementing different types of diode circuits and investigates the voltage and current relationships.
● To help them develop skills for calculating voltage gain, current gain, and overall gain in multistage BJT, JFET and MOSFET amplifiers.
● To provide the students with the capability of implementing different real-life analog electronic circuits.

Course Contents:

Lab 1-2: To familiarize students with electronic devices and Laboratory Equipment.
Lab 3: To study of V-I Characteristics curve of the P-N junction diode.
Lab 4-5: To study of Half-Wave Rectification circuit with and without filter.
Lab 6-7: To study of Full-Wave Rectification circuit (Bridge & Center-tapped) with and without filter.
Lab 8-9: To study clipping and clamping circuits.
Lab 10-11: To study BJT, JFET, and MOSFET characteristics.
Lab 12: Speech/ Audio amplification using NPN/PNP Transistor.
Lab 13: MOSFET as an amplifier and switch.
Lab 14: Different operational amplifier circuits.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO1	Explain the operation of diodes, calculation of operating point and perform DC biasing.
CO2	Design of the rectifier, clipping and clamping circuits.
CO3	Distinguish and interpret the operation of BJT, JFET and MOSFET amplifier circuits.
CO4	Differentiate between BJT and MOSFET switching circuits.
CO 5	Demonstrate team-based communication skills, magnify their moral standards, and apply these in practical life.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1				3								
CO 2			2	3								
CO 3			2	3								
CO 4			2	3								
CO 5									3	2	1	

Recommended Books

1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku

2. Introductory Circuit Analysis by Robert L.Boylestad
3. Electronic Devices and Circuit Theory by Robert L. Boylestad and Louis Nashlesky
4. Microelectronic Circuits- Sedra/Smith

Course Title: Engineering Graphics	**Credits:** 1.5
Course Code: 0732 1206D	**Contact hours:** 3 hours/week

Rationale of the Course:
This course is a practical application of knowledge pertaining to all geometric features of a whole product, or assembly or sub-assemblies. Pictorial presentation by means of geometric shapes, lines, and dimensions is mandatory for engineering students. All engineering students need to have basic engineering graphics knowledge to express their thoughts and ideas.

Course Objectives:
The objectives of this course are to:

- provide the students with necessary skill to read, understand, and create mechanical engineering drawing
- familiarize the students to acquire and use engineering drawing skills on creating accurate, clear sketches of different mechanical objects following the information and instructions
- make students able to draw different types of angle projections, orthographic views, auxiliary, sectional views, isometric views, etc.
- enable students to acquire knowledge required for advanced study of engineering drawing
- apply the drawing and drafting skills as problem-solving tools to resolve the primary design issues.

Course Content:
Introduction, Instruments and their uses, First angle and third angle projections, Orthographic drawing, Sectional views. Isometric views, Missing lines and views.

Course Learning Outcomes, COs
Upon successful completion of this course, student will be able to:

CO 1: explain basic concepts of engineering drawing as an important form of conveying technical information;

CO 2: apply principles of engineering visualization and projection theory to prepare mechanical engineering drawings (2D and 3D), using conventional and modern drawing tools;

CO 3: practice drawing orthographic projection, sectional views, and isometric views of different mechanical parts;

CO 4: create a review report on mechanical components drawing using the engineering drawing-specific knowledge and skill for the multidisciplinary design team comprised of engineering professionals.

Mapping of COs with POs

ঈড়ঁৎংব খবধৎহরহম ঙঁঃপড়সবং (ঈঙ)	ঋঁহফধসবহঃধষ ঝশরষষং				ঝড়পরধষ ঝশরষষং			ঞযরহশরহম ঝশরষষং		চবৎংড়হধষ ঝশরষষং		
	চঙ১	চঙ২	চঙ৩	চঙ৫	চঙ৬	চঙ৯	চঙ১০	চঙ৪	চঙ৭	চঙ৮	চঙ১১	চঙ১২
ঈঙ ১	১			৩								
ঈঙ ২	২	১										
ঈঙ ৩	১			২								
ঈঙ ৪				২		১						

Mapping Course Learning Outcomes (COs) with the Teaching-Learning & Assessment Strategy

ঈড়ঁৎংব খবধৎহরহম ঙঁঃপড়সবং (ঈঙং)	ঞবধপযরহম-খবধৎহরহম ঝঃৎধঃবমু	অংংবংংসবহঃ ঝঃৎধঃবমু
ঈঙ ১	খবপঃঁৎব ʼংরহম নড়ধৎফ ধহফ ওহংঃৎঁপঃরড়হ সধঃবৎরধষং	ছঁরু ধহফ ঝবসবংঃবৎ-বহফ ড়ৎধষ বীধসরহধঃরড়হ
ঈঙ ২	খবপঃঁৎব ʼংরহম নড়ধৎফ ধহফ ঞঁঃড়ৎরধষ	ছঁরু, উৎধরিহম ধংংবংংসবহঃ, ধহফ ঝবসবংঃবৎ-বহফ ড়ৎধষ বীধসরহধঃরড়হ
ঈঙ ৩	খবপঃঁৎব ʼংরহম নড়ধৎফ ধহফ ঞঁঃড়ৎরধষ	ছঁরু, উৎধরিহম ধংংবংংসবহঃ, ধহফ ঝবসবংঃবৎ-বহফ ড়ৎধষ বীধসরহধঃরড়হ
ঈঙ ৪	খবপঃঁৎব ʼংরহম নড়ধৎফ ধহফ অংংরমহসবহঃ	ছঁরু ধহফ ঝবসবংঃবৎ-বহফ ড়ৎধষ বীধসরহধঃরড়হ

Books Recommended:

1. K.V. Reddy, Textbook of Engineering Drawing, BS Publications, India.
2. K. Rathnam, A First Course in Engineering Drawing, Springer Nature Singapore Pte Ltd.
3. M.B. Shah and B. C. Rana, Engineering Drawing, Dorling Kindersley (India) Pvt Ltd.
4. Colin H. Simmons and Denis E. Maguire, Manual of Engineering Drawing to British and International standards, 2e, Elsevier Newnes, Oxford.
5. K. Morling, Geometric and Engineering Drawing, Elsevier Ltd. USA.

Course Title: Basic Physics Lab	**Credits:** 1.5
Course Code: 0533 1202D	**Contact hours:** 3 hours/week

Rationale

A Physics lab aids students in establishing the relevance of the theory. It brings clarity to the mind of the students regarding the fundamental concept of the subject.

Course Objectives:

1. To familiarize the students with physics experimentation.
2. To have first-hand laboratory experience in verifying different laws of physics.
3. To orient students with scientific reporting.

Course Content:

Mechanics:

(i) Determination of the moment of inertia of a flywheel.
(ii) Determination of "g" by and moment of inertia of a compound pendulum.
(iii) Determination of "g" by and moment of inertia of a Kater's pendulum.
(iv) Using a flat spiral spring: a) Verification of Hooke's Law and determination of stiffness constant; b) Determination of "g" and the effective mass of the spring; c) Determination of modulus of rigidity of the material of the spring.

Properties of Matter:

(i) Determination of Young's Modulus by the method of bending.
(ii) Determination of Rigidity Modulus of a wire by Static method.

Electricity:

(i) Determination of galvanometer resistance by half deflection method.
(ii) Determination of specific resistance of a wire by Wheatstone's bridge (with end correction).
(iii) Measurement of low resistance by the method of fall of potential.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO 1	investigate the laws and principles of simple harmonic oscillation, conservation of energy, etc.
CO 2	measure the material properties like Young's modulus and modulus of rigidity.
CO 3	apply the knowledge of electric circuits, underlying physics of the resistance, galvanometer, Wheatstone's bridge, etc. in designing experiments to measure observable quantities.
CO 4	collect and analyze experimental data, and draw logical conclusions from the data.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3											
CO 2				2								
CO 3									2			
CO 4										2		

Recommended Books

1. Topping: Errors of Observation and Their Treatment
2. Worsnop, B.L. and Flint, H. T.: *Advanced Practical Physics*
3. Chowdhury, S. A. and Basak, A. K.: *Byaboharik PadarthaBidya*
4. Ahmed, G. and Uddin, M.S.: *Practical Physics*

Second Year First Semester

Course Title: Principles of Economics	**Credits:** 3.0
Course Code: 0311 2105D	**Contact hours:** 3 hours/week

Rationale

This course offers a framework for comprehending how people, organizations, and governments make decisions in a world with scarce resources. It helps in illuminating the trade-offs that individuals must make as well as the motivating factors behind choices. It covers the basic knowledge of microeconomics as well as macroeconomics. The microeconomics topics include basic principles of economics, demand, supply, and market mechanism, consumer and producer theory. In addition, it emphasizes various macroeconomic measurements and the macroeconomic relationships within an economy. It also aims to introduce macroeconomic analysis outlining how the national income is measured and determined. It also provides a framework in which the interaction of money and goods and services markets can be developed, allowing students to understand the process by which the levels of economic activity and employment are determined. In addition, economic development provides students with an understanding of economic theories and analysis in the field of development economics.

Course Objectives

1. To facilitate fundamental concepts of economics, such as scarcity, market mechanism, utility, budget line and their role in optimizing consumption.
2. To develop ability to compare different kinds of production processes, cost structures, and markets.
3. To provide the basic concept on the determination of national income, inflation, and rate of unemployment.
4. To familiarize the students with the mechanism of forming monetary and fiscal policies.
5. To explainhuman development index, measurements of poverty and inequality, Sustainable Development Goals (SDG).

Course Contents:

Introduction to Economics: Definition of economics and scope, economic resources and their allocation problems-scarcity, choice, opportunity cost, efficiency using production possibilities frontier, market vs planned economy, organization of market economy using circular flow diagram. **Theory of Market:** Concept of demand, supply and market equilibrium; determinants of demand and supply; shifting of demand and supply curves; price elasticity of demand and supply, application of elasticities; effect of government policies, i.e., tax, subsidy, price ceiling, flooring, rationing in the market. **Theory of Consumer Choice:** Concepts of utility; law of diminishing marginal utility; indifference curve; budget constraint; consumer's equilibrium. **Theory of Firm:** (a) Production- Production function; law of diminishing return; production and employment decision of firm; (b) Costs-accounting and economic cost, fixed and variable costs, total, average and marginal costs; (c) Industrial organizations of firms-characteristics of perfect competition, monopoly, oligopoly, monopolistic competition; determination of profit. **National Income Measurement and Determination:** GDP, GNI and personal disposable income; various methods of GDP measurement and their shortcomings; real vs. nominal GDP; growth rate and business cycle; Keynesian multiplier model of national income determination. **Inflation and Unemployment:** Definition ofinflation and unemployment; computation of inflation and unemployment rates; types, causes and consequences of inflation and unemployment; remedial measures, Phillips Curve. **Money, Banking and Monetary Policy:** Definition and functions of money; components of money supply and money demand; commercial banks and money creation process; central bank and instruments of monetary policy. **Budget, Fiscal Policy and Development:** Definition, objectives, types and instruments of fiscal policy; budget, types of taxation; heads of government expenditure; ADP; growth vs development; human development index; measurements of poverty and inequality;Sustainable Development Goals (SDG).

Course Learning Outcomes (COs)

After the successful completion of the course, students will be able to:

CO1. Describe scarcity, opportunity cost and Production Possibility Frontier.

CO2. Determine market equilibrium, consumer equilibrium, firm's profit, and national income.

CO3. Evaluate overall macroeconomic performance related to economic growth and inflation.

CO4. Analyse economic issues with graphs and numbers.

CO5. Develop critical thinking ability by relating economic policies with real-world scenarios.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2											
CO 2	2											
CO 3				2				2		3		
CO 4					3	3						
CO 5				3	3			2		2		

Recommended Books

1. Arnold, R. A. (2018). Economics, Southwestern Publishing Company, 13th Edition
2. Colander, D. (2019). Economics,McGraw-Hill Education; 11th edition
3. Hubbard G. and O'Brien A. (2022). Economics,Pearson; 8th edition

4. Mankiw, N. G. (2021). Principles of Economics, Thomson Southwestern Publishing, 9th Edition
5. Samuelson, P. A. and Nordhaus. D. (2009). Economics, McGraw-Hill USA, 19th Edition.

Course Title: Computer Aided Drawing	**Credits:** 1.5
Course Code: 0715 2108D	**Contact hours:** 3 hours/week

Rationale of the Course:
Computer Aided Drawing (CAD) is an introductory course entailing the running/operation of a typical CAD system. This course deals with 2D computer-aided-drawings a designed product for manufacturing applications.

Course Objectives:
The objectives of this course are to:

- introduce computer aided drawing/drafting (CAD) system using the current version of the AutoCAD software
- teach the use of modern CAD software to produce two-dimensional drawings / computer models of a designed product for manufacturing applications
- provide the students with the basic understanding of AutoCAD commands and capabilities by providing hands on training environment
- help to learn the basic features and tools used to create and modify 2D drawings and proper dimensioning of a designed product
- facilitate to create a drawing sheet and plot 2D drawings.

Course Contents:
Introduction to Computer Aided Design and Drafting (CADD), Introducing the AutoCAD window, Opening a drawing, Using commands, Specifying coordinates, Setting up a drawing, Drawing in two dimensions (2D), Viewing the drawing, Editing the drawing: Basic and advance tools, Organizing drawing with layers, colors and line types, Drawing dimensions, Adding text to drawing, Storing and linking data with graphics, Getting and exchanging data from drawing.

Course Learning Outcomes, COs
Upon successful completion of this course, the students will be able to:

CO 1: demonstrate the basic commands, tools and concepts of AutoCAD software for creating an engineering drawing with the use of computers;
CO 2: practice two-dimensional (2D) drawings /computer models of a designed product for manufacturing applications using AutoCAD software;
CO 3: manipulate drawings through modifying/editing tools and plotting techniques, and produce template drawings;
CO 4: develop mid-level CAD user skills using the current version of the AutoCAD software;
CO 5: apply the concepts and technical skills learned in the course to real-world design problems and current professional practices.

Mapping of COs with POs

Course Learning Outcomes (CO)	Fundamental Skills				Social Skills			Thinking Skills		Personal Skills		
	PO1	PO2	PO3	PO5	PO6	PO9	PO10	PO4	PO7	PO8	PO11	PO12
CO 1	1			3								
CO 2	2	1										
CO 3	1			2								
CO 4				2		1						
CO 5				2		2						2

Mapping Course Learning Outcomes (COs) with the Teaching-Learning & Assessment Strategy

Course Learning Outcomes (COs)	Teaching-Learning Strategy	Assessment Strategy
CO 1	Presentation with projectors and instruction materials	Drawing assignment, oral exam, semester-end exam
CO 2	Lecture using projectors, Tutorial and Assignment	Drawing assignment, oral exam, semester-end exam
CO 3	Lecture projectors, Tutorial and Assignment	Drawing assignment, oral exam, and semester-end exam.
CO 4	Lecture using projectors and tutorial	Drawing assignment, oral exam and group project
CO 5	Lecture using PC softwares and tutorial	Semester-end Oral examination and individual project

Books Recommended:

1. Up and Running with AutoCAD 2017: 2D and 3D Drawing and Modeling – By Gindis, Elliot, Academic Press Publisher
2. Introduction to AutoCAD 2011: 2D and 3D Design – By Yarwood, Alf, Taylor & Francis Publisher
3. Beginning AutoCAD 2011: Exercise Workbook – By Shrock, Cheryl R., Industrial Press.
4. Tutorial Guide to AutoCAD 2020: 2D Drawing, 2D Modeling – By Lockhart, Shawna, SDC Publications.

5. AutoCAD 2016 Tutorial: First Level 2D Fundamentals – By Shih, Randy H., SDC Publications.
6. AutoCAD 2004 Bible – By Finkelstein, Ellen, Wiley Publisher.

Course Title: Cost and Management Accounting		**Credits:** 3.0
Course Code: 0411 2105D	**Contact Hours:** 3 hours/week	

Rationale
This course will enable the students to acquire a conceptual knowledge on cost accounting and management accounting.

Course Objectives
(i) describe the cost concepts, cost behavior , and cost accounting techniques that are applied to manufacturing and service businesses.
(ii) interpret cost accounting statements;
(iii) provide the students with the capability to apply theoretical knowledge in decision making.
(iv) Analyse and evaluate information for cost ascertainment, planning, control of business operations
(v) discuss the various techniques available to measure managerial performance and to motivate employees toward organizational goals.

Course Contents:
Introduction to Cost Accounting: Definition of Cost Accounting, Comparison of Cost Accounting and Financial Accounting; The role of Cost Accounting; Methods and Techniques of Cost Accounting; Characteristics of an Ideal Cost Accounting System, **Cost Concepts, Classifications and Statements:** Cost Object; Expenditures, Cost, Expense and Loss; Cost Classifications; Cost Data and Uses; The Chart of Accounts; Statement of Cost of Goods Manufactured and Sold; Cost Statement or Cost Sheet, **Costing and Control of Materials:** Classification of Materials; Accounting for Materials; Store ledger(FIFO & WAM) method; Inventory Planning; Ordering Cost, Holding Cost and EOQ; Effect of Quantity Discounts on EOQ; Safety Stock and Reorder Point; Material Control Methods; Materials Requirement Planning System. Practical problem. **Costing and Control of Labour**: Productivity and Labour Costs; Costs included in Labour; Accounting for Labour; Time Keeping, Computation of total payroll and Allocation of Payroll costs; Different incentive plan; Labour cost Control, Labour Turnover and Control of Labour Turnover; Learning Curve Theory. Practical problem & solution, **Costing and Control of Manufacturing Overhead**: Manufacturing Overhead Costs; Actual Vs. Normal Costing of Manufacturing Overhead; Production Capacity, Predetermined Overhead Rates; Departmental vs. Plant-wise Overhead Rates; Separating Mixed Costs. Scatter-graph; High-low Method and Regression Analysis; Accounting for Manufacturing Overhead; Analysis and Disposition of Under-applied-and Over-applied Overhead, **Contract Costing**: Determination of profit of completed and incomplete contracts, **Introduction of Management Accounting** :Definition-process of Management Accounting, characteristics of Management Accounting, scope of Management Accounting, purpose and objectives of Management Accounting, Comparison of Management Accounting and Financial Accounting, **Cost Terms, Concepts and Classifications:** Cost Behaviour (Analysis and Use): General cost classifications- product costs versus period costs- cost classifications on Financial Statements. Types of cost behaviour patterns- the Analysis of Mixed Costs, High-low method, **Cost-Volume-Profit Relationships:** The basics of CVP analysis- Break -even analysis- Break-even chart- Sales Mix. Business application and mathematical problem of CVP analysis, **Budget:** Define Budget, Types of Budget, Cash budget, purchase budget, sales budget, flexible budget and Related problems, **Standard Costing:** Meaning and Objectives- Types of ratios. Standard Costing and its uses for making business decision. Variance calculation, Decision making process from these calculations, **Standard Costing:** Variance calculation, Decision making process from these calculations.

Course Learning Outcomes:
After the successful completion of the course, the student will be able to-

CO1	understand the theories and concepts of costing needed to cope up with future business challenges.
CO2	make technically of the analysing of different costing method which is preferable for every stakeholder.
CO3	Analyse of both qualitative and quantitative data to formulate best costing methods
CO4	identify and Analyse both qualitative and quantitative standards to formulate best control methods
CO5	able to interpret the techniques and methods used cost control and help to managers in decision making.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	1							2			
CO 2			3									
CO 3	3			1	1	2						

CO 4	2	3			3		3	2				
CO 5	2			2					2	3		

Recommended Books:

1. Cost Accounting –Volume-1 by Basu and Das;
2. Managerial Accounting by Ray H. Garrison, Eric W. Noreen
3. Cost Accounting by MutzUzry et al.

Course Title: Statistics & Probability	**Credits:** 3.0
Course Code: 0542 2102D	**Contact hours:** 3 hours/week

Rationale:

Acquiring knowledge on the statistical tools and techniques for exploring and analyzing the data.

Course Objectives

- Provide the knowledge on fundamental concepts of statistical methods
- Acquaint students with the basic tools of exploratory data analysis,
- Facilitate necessary knowledge about bivariate data analysis
- Make students understand the basic concepts of probability and probability distribution,
- Help the students conceptualize basic theories in Stochastic processes including Markov chain and queuing theory

Content of Course

Frequency distribution of data: population and sample. Collection and representation of statistical data. Tabulation of data. Class intervals. Frequency distribution, discrete, continuous and cumulative distributions. Histograms and frequency polygons. Graphical representation of data. **Statistical measures:** measures of central tendency - arithmetic mean, median, mode, geometric mean, weighted average, harmonic mean. Measures of dispersion - range, standard deviation, variance, coefficient of variation, moments, skewness, kurtosis. **Correlation theory:** linear correlation. Measures of correlation and its significance. Regression and curve fitting: linear and non-linear regression. Methods of least squares. Curve fitting. **Probability:** definition of probability and related concepts. Laws of probability. Discrete and continuous random variables. Mathematical expectations. Conditional probability. **Probability distributions:** binomial, Poisson and normal distributions and their properties. Stochastic process.

Course Learning Outcomes

After the successful completion of the course, the student will be able to-

CO 1	Explain basic concepts of statistics and describe various statistical tools
CO 2	Construct frequency distribution and present data graphically
CO 3	Compute and interpret different measures of central tendency, location, dispersion, and shape characteristics
CO 4	Analyze bivariate data and interpret the results
CO 5	Calculate probability of an event and derive probability distribution of a random variable

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			3								
CO 2	3			3								
CO 3	3			3								
CO 4	3			3								
CO 5	3			3								

Text Books

1. Devore J., (2009), Probability and Statistics for Engineering and the Sciences, 8th Edition, Brooks/Cole, Cengage Learning, California
2. Montgomery, D.C., Runger, G.C., (2003), Applied Statistics and Probability for Engineers, 3rd Edition, John Wiley & Sons, Inc., NY
3. Ross, S.M., (2007), Introduction to Probability Models, 9th Edition, Academic Press, NY

Reference Books:

1. Barlow R J, Statistics: A Guide to the Use of Statistical Methods in the Physical Sciences, Wiley, NY
2. Chisholm J S R & Morris R M, Mathematical Methods in Physics, North Holland
3. Hoel P G, Elementary Statistics, 3rd Ed, John Wiley, NY
4. Loveday R, Practical Statistics and Probability, Cambridge University Press, London
5. Mostafa M G, Methods of Statistics, Bangladesh

Second Year Second Semester

Course Title: Complex Variables, Laplace Transform and Fourier Series	**Credits:** 3.0
Course Code: 0541 2204D	**Contact Hours:** 3 hours/week

Rationale
The course gives the students a sound knowledge of Fourier transforms along with Fourier integrals, Laplace Transformation and complex variables.

Course Objectives
- To facilitate with basic ideas about complex variables
- To provide knowledge on Laplace transformation
- To acquaint with Fourier series and analysis

Course Contents:
Complex Variables: Complex numbers and their properties; De Moivre's theorem and its application; locus problem; functions of a complex variable; limit and continuity of a function of complex variable; analytical functions; the Cauchy-Riemann equations; Cauchy's theorem; singularity and poles; residues; simple contour integration and their uses in solving boundary value problems. **Laplace Transformations:** Definition of Laplace transform; Laplace transform of different functions; first shift theorem; inverse Laplace transform; linearity property; use of first shift theorem and partial functions; Laplace transform of derivatives; Laplace transform of an integral; the Heaviside unit function; the unit impulse function; the second shift theorem; periodic functions; convolutions; solution of ordinary differential equations by Laplace transform. **Fourier Series:** Fourier series; convergence of Fourier series; Fourier analysis; Fourier transforms

Course Learning Outcomes:
After the successful completion of the course, the student will be able to-

CO 1	Define the complex number system, complex functions and integrals of complex functions
CO 2	Apply the results/theorems in complex analysis to complex valued functions
CO 3	Understand Laplace transforms
CO 4	Discuss about analytic function and how to check analyticity based on Cauchy – Riemann equation
CO 5	Represent periodic functions using Fourier series

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3			2								
CO 2				3								
CO 3		2										
CO 4				3								
CO 5										3		

Recommended Books
1. KK Kodaira: Introduction to Complex analysis
2. H Jaffreys and B Jaffreys: Methods of Mathematical Physics
3. Spiegel, M. R.: Laplace Transform
4. Khanna, M. L.: Laplace Transforms

Course Title: Digital Logic Design	**Credits:** 3.0
Course Code: 0714 2201D	**Contact hours:** 3 hours/week

Rationale
The main aim of this course is to provide practical knowledge of the principles and practices of digital systems, both at the device and circuit levels. The course covers practical experiments on the topics of digital electronics, including Number Theory, Boolean Algebra, Logic Circuits, Logic Minimization Techniques, Multiplexers, Adders, Flip-Flops, Counters, Registers, State Machines, Memory Circuits, Digital /Analog Conversion, Programmable Logic Circuits and Microcomputer Bus Architecture. Upon completion, students should be able to construct, analyze, verify, and troubleshoot digital circuits using appropriate techniques and test equipment.

Course Objectives:
- To make the students able to design combination and sequential circuits based on some specifications.
- To make the students understand the principles of different registers and counters.
- To make the students understand the operation and application of different types of memory and programmable logic devices.
- To familiarize students with different logic families.

Course Contents:
Logic Gates: AND, OR, NAND, NOR, X-OR, X-NOR, Circuit Design. **Flip flops:** SR, JK, D, Master Slave, Application, and Synchronization. **Logic Circuits:** Coder, Decoder, Mux, Demux. **Counters:** Synchronous, Asynchronous, Up/Down, Ripple, Cascading. **Registers:** Shift registers. **Memory Devices:** ROM, RAM, Static, Dynamic, Memory Operation. **Arithmetic Circuits:** Adder, Carry, Look Ahead, ALU. **PAL:** Micro-program Control, FPGA, HDLA. **Logic Families:** TTL, CMOS, ECL, Tristate.

Course Learning Outcomes:
After the successful completion of the course, the student will be able to-

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	3	2										
CO 2	3				2							
CO 3	3				2							
CO 4	3	2										

Textbook

1. Digital Logic Design by Morris Mano
2. Digital Systems by Ronald Tocci, Neal Widmer, Greg Moss
3. Digital Principles and Applications by Donald P Lech, Albert Paul Malvino and Goutam Saha

Course Title: Digital Logic Design Lab	**Credits:** 1.5
Course Code: 0714 2202D	**Contact hours:** 3 hours/week

Rationale

The main aim of this course is to provide practical knowledge of the principles and practices of digital systems, both at the device and circuit levels. The course covers practical experiments on the topics of digital electronics, including Number Theory, Boolean Algebra, Logic Circuits, Logic Minimization Techniques, Multiplexers, Adders, Flip-Flops, Counters, Registers, State Machines, Memory Circuits, Digital / Analog Conversion, Programmable Logic Circuits and Microcomputer Bus Architecture. Upon completion, students should be able to construct, analyze, verify, and troubleshoot digital circuits using appropriate techniques and test equipment.

Course Objectives

- To help the students design combination and sequential circuits based on some specifications.
- To help the students understand the principles of different types of registers and counters.
- To help the students understand the operation and application of different types of memory and programmable logic devices.
- To familiarize students with different logic families.

Course Contents:

Introduction: Familiarization with digital logic design laboratory and tools within the lab

Experiment 1: Verification of De Morgan's laws.
Experiment 2: Verification of logic gates.
Experiment 3: Verification of universal gates.
Experiment 4: Implementation of Boolean functions using basic gates and universal gates.

CO1	Design combinational and sequential circuits using logic gates.
CO2	Describe the principles of different types of counters and register.
CO3	Describe the operation and application of different memory and programmable logic device types.
CO4	Explain the basics of different logic families.

Experiment 5: Implementation of half adder using basic and universal gates.
Experiment 6: Implementation of multiplexer and demultiplexer using basic gates.
Experiment 7: Interfacing of seven-segment display and display decoder.
Experiment 8: Implementation of encoder and decoder.
Experiment 9: Verification of latch and flip-flop.
Experiment 10 Implementation of J-K flip-flop using S-R flip-flop.
Experiment 11: Study of 4-bit ripple counter.
Experiment 12: Study of the 4-bit synchronous counter.
Experiment 13: Study of a 4-bit shift register.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CO1	Implement combinational and sequential circuits using the logic gate.
CO2	Demonstrate proficiency in using laboratory tools to carry out the experiments.
CO3	Conduct experiments to correlate the theoretical knowledge.
CO4	Demonstrate the ability to work as a team.
CO 5	Interface different IC using their specification sheet.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2			3								
CO 2				3	2							
CO 3				3								
CO 4									3	2	1	
CO 5					2							

Recommended Books

1. Digital Logic Design by Morris Mano
2. Digital Systems by Ronald Tocci, Neal Widmer, Greg Moss
3. Digital Principles and Applications by Donald P Lech, Albert Paul Malvino and Goutam Saha

List of Non-Major Courses
(offered by CSE for other departments)

Course Code	Course Title	Cr.
0611 xx00*	Introduction to Computing Applications Lab	3.0
0612 xx15*	Database Management and Programming	2.0
0612 xx16*	Database Management and Programming Lab	3.0
0612 xx17*	Computer Networking	3.0
0612 xx18*	Computer Networking Lab	1.5
0613 xx01*	Introduction to Computer Language	2.0
0613 xx02*	Introduction to Computer Language Lab	2.0
0613 xx03*	Introduction to Programming with Python	2.0
0613 xx04*	Introduction to Programming with Python Lab	2.0
0613 xx06*	Python Programming Lab	3.0
0613 xx11*	Data Structure	3.0
0613 xx12*	Data Structure Lab	1.5
0613 xx13*	Algorithm Design and Analysis	3.0
0613 xx14*	Algorithm Design and Analysis Lab	1.5
0714 xx19*	Computer Architecture	3.0

xx = 11/12/21/22/31/32
* (A-ARC, B-CEP, C-CEE, D-CSE, E-EEE, F-FET, G-IPE, H-PME, I-BMB, J-GEB, K-CHE, L-GEE, M-MAT, N-PHY, O-STA, P-FES, Q-MEE, W-SWE, a-ANP, b-BNG, c-ECO, d-ENG, e-PSS, f-PAD, g-SCW, h- SOC, i-BUS)

Detailed Syllabus
Non-Major Courses
(offered by CSE for other departments)

Course Title: Introduction to Computing Applications Lab	**Credits:** 3.0
Course Code: 0611 xx00*	**Contact Hours:** 6 hours/week

Rationale
This is a dedicated lab course that aims to make students familiar with the basic computing applications that are frequently used in real life. It covers preparing and working with spreadsheets, formatting spreadsheets, creating charts, using tools, and printing worksheets. It also covers the basics of databases, table designs, indexing, creating queries, creating forms, and printing reports. As part of statistical analysis, it also includes the introduction of statistical analysis, operation commands, data definition and manipulation commands, and procedure commands. All of these will help students to have a good understanding and hands-on experience with the abovementioned basic applications.

Course Objectives
- To assist students in developing a good understanding of the vastly used basic computing applications.
- To help students understand the basic concepts of spreadsheets, databases, and statistical analysis.
- To facilitate the basic knowledge about the workflows of the applications like Microsoft Excel/Access, SPSS.
- To assist students in developing their hands-on skills on the mentioned applications.
- To make students capable of applying their knowledge on the mentioned areas in real life.

Course Contents:
Spreadsheet Analysis: Introduction (Spreadsheet & its Applications, Menus & Toolbars), Working with Spreadsheets (Converting files to different formats, Importing, Exporting, Spreadsheet addressing, Computing data, Mathematical operations, Using formulas), Formatting Spreadsheets (Border & shading, Highlighting values, Visibility, Sorting, Filtering, Validation, Consolidation, Subtotal), Creating Charts (Selecting charts, Formatting charts, label, scaling, etc.), Using Tools (Error Checking, Spell Checks, Macros), Printing worksheet. **Database Applications:** Introduction (Database concepts, Tables, Queries, Forms, Reports), Working with Databases (Creating Tables, Table Design, Indexing, Entering data,

importing data), Creating Queries (SQL statements, Setting relationship, using wizards), Creating Forms, Creating & printing reports. **Statistical Analysis:** Introduction, Operation commands, Data definition, manipulation commands, and procedure commands like LIST, DESCRIPTIVES, FREQUENCIES, CROSSTABS, T-TEST, ANOVA, REGRESS, etc.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Prepare a working spreadsheet with all necessary functionalities.
CO 2	Explain the concepts of database (tables, queries, mapping, indexing, forms, etc.)
CO 3	Create a working database with tables and relations and make queries in the database.
CO 4	Analyze a large amount of statistical data and generate analysis reports using applications like SPSS.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Course Title: Database Management and Programming	**Credits:** 2.0
Course Code: 0612 xx15*	**Contact Hours:** 2 hours/week

Rationale:

This course is designed to introduce the basic concepts of database, learn the foundations of database systems, focusing on basics such as the relational algebra and data model, schema normalization, query optimization, and transactions.

Objectives:

- To facilitate knowledge about different issues involved in the design and implementation of a database system.
- To make students understand physical and logical database designs, database modeling, relational, hierarchical and network models
- To provide the knowledge of data manipulation language to query, update, and manage a database
- To help to develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, Client/Server (Database Server), Data Warehousing.

- To help apply the knowledge of design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS

.

Course Contents:

Introduction: Purpose of Database Systems, Data Abstraction, Data Models, Instances and Schemes, Data Independence, Data Definition Language, Data Manipulation Language, Database Manager, Database administrator, Database Users, Overall System Structure, Advantages and Disadvantage of a Database Systems. Data Mining and analysis, Database Architecture, History of Database Systems. **Relationship Entity-Model:** Entities and Entity Sets, Relationships and Relationship Sets, Attributes, Composite and Multivalued Attributes, Mapping Constraints, Keys, Entity-Relationship Diagram, Reducing of E-R Diagram to Tables, Specialization, Generalization, Attribute Inheritance, Aggregation, Alternative E-R Notations, Design of an E-R Database Scheme. **Relational Model:** Structure of Relational Database, Fundamental Relational Algebra Operations, the Tuple Relational Calculus, the Domain Relational Calculus, Modifying the Database. **Relational Commercial Language:** SQL, Basic structure of SQL Queries, Query-by-Example, Nested Sub queries, Complex queries, Integrity Constraints, Authorization, Dynamic SQL, Recursive Queries. **Relational Database Design:** Pitfalls in Relational Database Design, Functional Dependency Theory, Normalization using Functional Dependencies, Normalization using Multivalued Dependencies, Normalization using join Dependencies, Database Design Process. **File and System Structure:** Overall System Structure, Physical Storage Media, File Organization, RAID, Organization of Records into Blocks, Sequential Files, Mapping Relational Data to Files, Data Dictionary Storage, Buffer Management. **Indexing and Hashing:** Basic Concepts, Ordered Indices, B+ -Tree Index Files, B-Tree Index Files, Static and Dynamic Hash Function, Comparison of Indexing and Hashing, Index Definition in SQL, Multiple Key Access. **Concurrency Control:** Schedules, Testing for Serializability, Lock-Based Protocols, Timestamp-Based Protocols, Validation Techniques, Multiple Granularity, Multiversion Schemes, Insert and Delete Operations, Deadlock Handling. **Distributed Database:** Structure of Distributed Databases, Trade-off in Distributing the Database, Design of Distributed Database, Transparency and Autonomy, Distributed Query Processing, Recovery in Distributed Systems, Commit Protocols, Concurrency Control.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Describe the basic concepts and appreciate the applications of database systems

CO2	Illustrate the basics of SQL and construct queries using SQL.
CO3	Be familiar with a commercial relational database system by writing SQL using the system.
CO4	Be familiar with the relational database theory and be able to write relational algebra expressions for queries.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Database System Concept, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fourth edition
2. Files and Databases- An Introduction, Peter D. Smith and G.M. Barnes, Addison- Wesley
3. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, Third edition

Course Title: Database Management and Programming Lab		**Credits:** 3.0
Course Code: 0612 xx16*	**Contact Hours:** 6 hours/week	

Rationale:

Database System LAB course will concentrate on the design and implementation of a database system and applying SQL query.

Objectives:

- To introduce the basic concepts of database.
- Developing a real world database application.
- To learn the design of a database starting from the conceptual design
- To the implementation of database schemas and user interfaces to a database.

Course Contents:

Introduction: MySQL, Oracle, SQL, Data types, SQL / PLSQL, Oracle Software Installation, User Type, Creating User , Granting. **Basic Parts of Speech in SQL:** Creating Newspaper Table, Select Command (Where, order by), Creating View, Getting Text Information & Changing it, Concatenation, Cut & paste string (RPAD, LPAD, TRIM, LTRIM, RTRIM, LOWER, UPPER, INIT, LENGTH, SUBSTR, INSTR, SOUNDEX). **Playing The Numbers:** Addition, Subtraction, Multiplication, Division, NVL, ABS, Floor, MOD, Power, SQRT, EXR, LN, LOG, ROUND, AVG, MAX, MIN, COUNT, SUM, Distinct, SUBQUERY FOR MAX, MIN. **Grouping things together:** Group By, Having, Order By, Views Renaming Columns with Aliases. **When one query depends upon another:** Union, Intersect, Minus, Not in, Not Exists. **Changing Data:** INSERT, UPDATE, MERGE, DELETE, ROLLBACK, AUTOCOMMIT, COMMIT, SAVEPOINTS, MULTI TABLE INSERT, DELETE, UPDATE, MERGE. **Creating And Altering tables & views:** Altering table, Dropping table, Creating view, Creating a table from a table. **By What Authority:** Creating User, Granting User, Password Management. **An Introduction to PL/SQL**: Implement a few problems using PL/SQL (e.g. Prime Number, Factorial, Calculating Area of Circle, etc.). **An Introduction to Trigger and Procedure:** Implement few problems using Trigger and Procedures. **An Introduction to Indexing:** Implement indexing using a large database and observe the difference of Indexed and Non-Indexed databases.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Demonstrate the knowledge in projects with a commercial relational database system (Oracle) and design a team-based project.
CO2	Utilize the database design principles, SQL and PL SQL.
CO3	Demonstrate the relational database theory and be able to write relational algebra expressions for queries.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Database System Concept, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Fourth edition
2. Files and Databases- An Introduction, Peter D. Smith and G.M. Barnes, Addison- Wesley
3. Database Management Systems, Raghu Ramakrishnan and Johannes Gehrke, Third edition

Course Title: Computer Networking		**Credits:** 3.0
Course Code: 0612 xx17*	**Contact Hours:** 3 hours/week	

Rationale:

The aim of this course is to introduce key concepts and principles of computer networks to provide a solid understanding of the technologies that support modern networked computer systems. The course will use a top-down approach to study the Internet and its protocol stack. Instances of architecture, protocol, application - examples will include email, web and media-streaming. It will cover communications services (e.g., TCP/IP) required to support such network applications. The implementation and deployment of communications services in wired and wireless LAN environments will be followed by a discussion of issues of network-security and network-management. Throughout the course, the Internet's architecture and protocols will be used as the primary examples to illustrate the fundamental principles of computer networking.

Objectives:

- To provide basic knowledge about various network technologies and techniques
- To facilitate idea about the importance of layering, and the OSI reference model
- To provide knowledge of understanding of the design and operation of an IP network, such as the Internet, and explain the purpose and function of its various components
- To make them understand the general principles behind addressing, routing, reliable transmission and other stateful protocols as well as specific examples of each
- To make them be able to describe the issues in connecting heterogeneous networks

Course Contents:

Introduction: Introduction to Computer Networks, Network Goals, Applications of Networks, Network Structure, Network Architectures, The OSI Reference Model, Data Transmission in the OSI Model, OSI Terminology, The ARPANET. **Local Area Network:** LAN Technology - Architecture, Topology. **Wired LANs:** Ethernet and Fast-Ethernet, Token Ring, FDDI. **Wireless LANs:** IEEE 802.11, Bluetooth. Backbone Networks, Virtual LANs. **Wide Area Network:** SONET, Virtual Circuit Networks - Frame Relay, ATM and ATM LANs. **Network Layer:** Logical Addressing.**Internet Protocol:** Internetworking, Routing Protocol, IPv4 and IPv6. **Address Mapping, Error Reporting and Multicasting:** ICMP, IGMP, ICMv6. Delivery, Forwarding and Routing. **Transport Layer:** Process-to-Process delivery, Transport Services, Protocol mechanisms, TCP, UDP, SCTP, Congestion and QoS. **Application Layer:** Domain Name System, Abstract Syntax Notation One (ASN.1), Network Management - SNMPv2, Electronic mail - SMTP and MIME, Uniform Resource Locator (URL) and Universal Resource Identifier (URI), Hypertext Transfer Protocol (HTTP). **Wireless and Mobile Networking**: Wireless Networking: Issues and Trends, Wireless Physical Layer Concepts, Wireless Cellular Networks, Mobile IP - IPv4, IPv6, TCP over Wireless, Ad Hoc Networks: Issues and Routing, Wireless Sensor Networks, Wireless Mesh and Multi-Hop Relay Networks, Wireless Network Security, Energy Management in Ad Hoc Wireless Networks.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Explainthe architecture of a computer network and describe how each device in a network communicates with each other
CO 2	Identifythe basic network protocols in each layer of a TCP/IP stack and compare with its counterpart OSI layer
CO 3	Compare different LAN technologies and their basic principles
CO 4	Explain the use of subnetting and use the technique to divide a large network into smaller logical subnetworks
CO 5	Analyze routing protocols and algorithms
CO 6	Investigate transport layer services, multiplexing/demultiplexing and congestion control
CO 7	Identify and analyze different types of application layer protocols such as HTTP, FTP, POP3, SMTP and DNS.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Data Communications and Networking – Behrouz A. Forouzan
2. Computer networks – A. S. Tanenbaum, Addison-Wesley.
3. Computer Networking: a Top-down Approach – James F. Kurose, Keith W. Ross
4. Computer Networks: A Systems Approach – Peterson and Davie.

Course Title: Computer Networking Lab	**Credits:** 1.5
Course Code: 0612 xx18*	**Contact Hours:** 3 hours/week

Rationale:

The aim of this lab course is to provide practical knowledge in computer networks. Students will learn subnetting and will design a network using Packet Tracer and analysis the behavior of TCP/IP layers. The students will gain practical knowledge of configuring Switch, Router, DHCP, FTP servers.

Objectives:

- To help students designing and implementing VLSM addressing schemes in a heterogeneous computer network using Packet Tracer
- To make them capable to configure Switch, Router and other end devices
- To make them capable to configure DHCP, SMTP and FTP servers
- To make them capable to design and implement a system using socket programming

Course Contents:

Subnetting and designing a network using Packet Tracer. Analysis of the TCP/IP behavior. Exploring several aspects of different Application layer protocols such as HTTP and DNS. Packet analysis. Server configuration: DHCP, SMTP, FTP, Web Switch and Router Configuration. Socket Programming.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Design and implement a heterogeneous computer network
CO 2	Analyze the behavior of different Application and Transport layer protocols
CO 3	Configure switch, router and end devices in a networkand test connectivity
CO 4	Analyze and implement IP addressing and subnetting of IP networks
CO 5	Design a system using socket programming

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Data Communications and Networking – Behrouz A. Forouzan
2. Computer networks – A. S. Tanenbaum, Addison-Wesley.
3. Computer Networking: a Top-down Approach – James F. Kurose, Keith W. Ross
4. Computer Networks: A Systems Approach – Peterson and Davie.

Course Title: Introduction to Computer Language	**Credits:** 2.0

Course Code: 0613 xx01*	**Contact Hours:** 2 hours/week

Rationale:

To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To provide students a basic understanding of computer hardware and how a computer works
- To make students understand the basic terminology used in computer programming
- To facilitate with knowledge of how to write, compile and debug programs in the C language
- To help students write programs involving decision structures, loops, functions, and pointers
- To help students develop skills on standard programming practices and how to build up their logic and how to implement them.

Course Contents:

Computer Basics: Concept on Computer Hardware, Software, and its classification. **C-Language**: Preliminaries, Program constructs variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers and structures; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Define the basic terminologies related to computer and explain the workflow of a computer and computer program
CO 2	Implement the basic programming commands, control structures (if-else, loop), function, arrays, pointers, structures, unions, files and use them to write a computer program
CO 3	Build a computer program o to solve a specific programming problem and determine the errors in programs written by other programmers
CO 4	Apply best practices for code organization and maintainability in C programming languages.
CO 5	Be able to apply the knowledge and skills gained in this course to real-world programming tasks and projects.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction to Computer Language Lab		**Credits:** 2.0
Course Code: 0613 xx02*	**Contact Hours:** 4 hours/week	

Rationale:

To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To help to develop skills to work with C compilers and how to use run programs on the computer
- To foster the analytical and critical knowledge to build up logic and implement them using programming language C
  To facilitate necessary knowledge about how to design programs involving decision structures, loops, functions, and pointers
- To help to develop skills to debug codes by giving an in-depth idea about different syntax errors, exceptions and how to fix them

- To provide the knowledge of using basic searching and sorting algorithms

Course Contents:

Computer Basics: Concept on Computer Hardware, Software, and its classification. **C-Language:** Preliminaries, Program constructs variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers and structures; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library. **Problem Solving:** Basic Calculator, Odd/Even Test, Showing Letter Grade from Number, Drawing different shapes using Asterisks (*), GCD, Palindrome, Fibonacci Series, Geometric Mean, Quadratic Formula, Cumulative Sum, Cumulative Product, Weighted Average, Generating Prime Number using Sieve of Eratosthenes, Prime Factorization, Big Mod, SOD, NOD, Permutation, Combination, Finding Areas, Basic Geometry Problems, Factorial, Leap Year, Tower of Hanoi, String Manipulation: Vowel and Consonant Count, Reversing a Word, Matrix Multiplication, Piglatin Generator.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Define the basic terminologies related to computer and explain the workflow of a computer and computer program
CO 2	Implement the basic programming commands, control structures (if-else, loop), function, arrays, pointers, structures, unions, files and use them to write a computer program
CO 3	Build a computer program o to solve a specific programming problem and determine the errors in programs written by other programmers
CO 4	Apply best practices for code organization and maintainability in C programming languages.
CO 5	Be able to apply the knowledge and skills gained in this course to real-world programming tasks and projects.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction to Programming with Python		**Credits:** 2.0
Course Code: 0613 xx03*	**Contact Hours:** 2 hours/week	

Rationale

In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives
- help students conceptualize basic theories of computer programming
- make the students understand fundamental components of python programming
- develop skills for writing computer programs using all necessary branches of Python
- accumulate basic ideas about data structures and data manipulations.

Course Contents:

Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with json; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	implement knowledge of Python for writing computer programs;
CO 2	apply solutions to real-life problems using the necessary components of Python;
CO 3	identify errors from a program and use exception handlers to handle errors and exceptions;
CO 4	design basic data structures to solve efficient data storage issues;
CO 5	implement object oriented programming and modular concepts, etc., in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Introduction to Programming with Python Lab		**Credits:** 2.0
Course Code: 0613 xx04*	**Contact Hours:** 4 hours/week	

Rationale

In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives

- help students conceptualize basic theories of computer programming
- make the students understand fundamental components of python programming
- develop skills for writing computer programs using all necessary branches of Python
- accumulate basic ideas about data structures and data manipulations.

Course Contents:

Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with json; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	implement knowledge of Python for writing computer programs;
CO 2	apply solutions to real-life problems using the necessary components of Python;
CO 3	identify errors from a program and use exception handlers to handle errors and exceptions;
CO 4	design basic data structures to solve efficient data storage issues;
CO 5	implement object oriented programming and modular concepts, etc., in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook:

1. Learning Python, By Mark Lutz, 5thEdition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Python Programming Lab		**Credits:** 3.0
Course Code: 0613 xx06*	**Contact Hours:** 6 hours/week	

Rationale

In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives

- help students conceptualize basic theories of computer programming

- make the students understand fundamental components of python programming
- develop skills for writing computer programs using all necessary branches of Python
- accumulate basic ideas about data structures and data manipulations.

Course Contents:

Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with json; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	implement knowledge of Python for writing computer programs;
CO 2	apply solutions to real-life problems using the necessary components of Python;
CO 3	identify errors from a program and use exception handlers to handle errors and exceptions;
CO 4	design basic data structures to solve efficient data storage issues;
CO 5	implement object oriented programming and modular concepts, etc., in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook:

1. Learning Python, By Mark Lutz, 5thEdition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Data Structure	**Credits:** 3.0
Course Code: 0613 xx11*	**Contact Hours:** 3 hours/week

Rationale:

To provide the students with solid foundations in the basic concepts of programming, that is,in data structures and related algorithms.To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter in future and how to study their computational complexities.

Objectives:

- To explain the purpose and the mathematical background of algorithm analysis
- To facilitate necessary knowledge about the abstract data types,such as, stacks, queues and dequeues
- To familiarize with variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated
- To facilitate necessary knowledge about the characteristics and optimal behavior of hash tables for access and retrieval
- To provide the knowledge of various sorting algorithms and the run-time analysis required to determine their efficiencies

- To help them understand various tree traversal techniques and graph algorithms

Course Contents:

Internal Data Representation: Specification, representation. **Asymptotic analysis:** Recurrences. **Stack and Queue:** Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to post-fix expressions conversion and evaluation, balancing parentheses using stack, basic queue operations (enqueue, dequeue), circular queue/ dequeue, queue-class implementation using array and linked list, application- Josephus problem, Palindrome checker using stack and queue. **Sorting:** merge sort, quick sort (randomized quicksort), and distribution sort (counting sort, radix sort, bucket sort), lower bounds for sorting. **Linked List:** Singly Linked List, Doubly Link List, Traversal, Insertion, Deletion, and Modification. **Searching:** Application of Binary Search- finding elements in a sorted array, finding nth root of a real number, solving equations. **Heap:** Min-heap, max-heap, Fibonacci-heap, applications- priority queue, heap sort. **Set Operations & Disjoint Set:** Union finding, path compression. **Binary Tree (BIT):** Binary tree representation using array and pointers, traversal of Binary Tree (in-order, pre-order, and post-order). Binary Index tree, Segment tree, Range Minimum Query (RMQ), Improvement of BIT: AVL Tree. **Graph:** Graph representation (adjacency matrix/adjacency list), basic operations on graph (node/edge insertion and deletion), traversing a graph: Breadth-first search (BFS), Depth-first search (DFS), Topological Sort.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Define and explain the fundamental data structures such as lists, queues, trees, etc.
CO 2	Illustrate the concept of algorithm complexity analysis
CO 3	Compare tradeoffs in the design and implementations of the data structures
CO 4	Select appropriate algorithms to use in specific applications and apply
CO 5	Design data structures to store and manipulate data while solving real life problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Advanced Data Structures - Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms - Thomas H. Cormen , Charles E. Leiserson

Course Title: Data Structure Lab		**Credits:** 1.5
Course Code: 0613 xx12*	**Contact Hours:** 3 hours/week	

Rationale:

To provide the students with solid foundations for implementation of various linear and non-linear data structures. To teach the students how to select, design and develop data structures and algorithms that are appropriate for problems that they might encounter.

Objectives:

- To facilitate necessary hands-on knowledge to implement and manipulate various linear and non-linear data structures
- To facilitate necessary hands-on knowledge to implement various searching and sorting algorithms
- To facilitate necessary hands-on knowledge to design and develop real-world applications using suitable data structure.

Course Contents:

This course is based on the theory course 0613 xx12. This course has been design to facilitate students a hands-on experience to implement various types of linear and non-linear data structures and related algorithms to manipulate those data structures. Besides, students will implement various sorting and searching algorithms also. At the end of the course the students should be able to design and develop data structures and sorting and searching algorithms that are used in various real-world applications.

Here are the data structures and algorithms that will be implemented by the students.

- Binary and linear searching using arrays; representing stack and queue using arrays
- Different types of linked lists; representing stacks and queues using linked list
- Converting infix expression to postfix and evaluating postfix expressions
- Implementing few algorithms using recursion
- Implementing various sorting algorithms
- Creating, manipulating and traversing Binary and Ternary trees
- Creating, manipulating, traversing and searching BSTs

- Creating and manipulating max/min heaps and implementing merge sort
- Implementing Kruskal's minimum spanning tree algorithm using disjoint sets
- Implementing compression algorithm using Huffman Coding graph
- Creating, manipulating, traversing and searching various types of graphs
- Implementing few well-known graph based algorithms
- Implementing, Prim's algorithm for finding minimum spanning tree
- Creating, manipulating, traversing AVL trees;solving few string based problems using string operations.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Understand various data representation techniques in the real world.
CO 2	Implement linear and non-linear data structures.
CO 3	Analyze various algorithms based on their time and space complexity
CO 4	Design and develop real-world applications using suitable data structure.
CO 5	Identify suitable data structure to solve various computing problems

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PO of the corresponding department

Textbook

1. Advanced Data Structures, Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms Thomas H. Cormen, Charles E. Leiserson

Course Title: Algorithm Design and Analysis	**Credits:** 3.0
Course No.: CSE 207M	**Contact hours:** 3 hours/week

Rationale:

To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To familiarize with the asymptotic performance of algorithms
- To familiarize with rigorous correctness proofs for algorithms
- To demonstrate familiarity with major algorithms and data structures
- To facilitate with the necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Contents:

Analysis of Algorithm: Asymptotic analysis: Recurrences, Substitution method, Recurrence tree method, Master method, Divide and Conquer Algorithms. **Hash Table:** Hash tables, hash function, open addressing, perfect hashing, single and multi-probe hashing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, Huffman Coding. **Dynamic Programming:** Elements of DP (Optimal substructure, overlapping subproblem), Coin change-related problem, 0-1 knapsack, Longest Common Subsequence finding problems, LCS and LIS/LDS variations, Matrix Chain Multiplication. Red Black Tree and Binomial Heaps, Stassen's algorithm. **Graph**: Strongly Connected Components, Euler Path, Articulation Point, Bridge, Bi-connected Components, Dijkstra's Shortest Path Algorithm, Bellman-Ford algorithm, and negative cycle detection, Floyd-Warshall all pair shortest path algorithm, shortest path in Directed Acyclic Graph. **Minimum Spanning Tree (MST):** Prim's algorithm and Kruskal's algorithm. **Network Flow:** Flow Networks, Max-Flow Min-Cut Theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum Bipartite Matching, minimum path cover, minimum edge cover. **Matrix Operation:** Matrix Chain Multiplication. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. Traveling salesman problem. **Geometric algorithm:** Properties of Line-Segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. RSA public key generation, NP-Completeness, NP-hard and NP-complete problems. **String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree, and Suffix Array. Basic combinatorics, Probability, and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT, and FFT.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Analyze the computational and memory complexities of algorithms
CLO 2	Prove the correctness of algorithms
CLO 3	Explain how and why the algorithms work
CLO 4	Apply the algorithms to solve real-life problems
CLO 5	*Decide when to use which algorithm*
CLO 6	Synthesize algorithms to design complex solutions
CLO 7	Explain why one algorithm works better than others in different scenarios

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department.

Textbook
1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. 102 Combinatorial Problems byTituAndreescu&Zuming Feng
3. Problem-Solving Methods in Combinatoricsby Pablo Soberón
4. Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: Algorithm Design and Analysis Lab	**Credits:** 1.5
Course No.: CSE 208M	**Contact hours:** 3 hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:
- To familiarize with the asymptotic performance of algorithms
- To familiarize with rigorous correctness proofs for algorithms
- To demonstrate a familiarity with major algorithms and data structures
- To facilitate with necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Contents:

Hash Table: Hash tables, hash function, open addressing, perfect hashing, single and multi-probe hashing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, Huffman Coding. **Dynamic Programming:** Elements of DP (Optimal substructure, overlapping sub problem), Coin change related problem, 0-1 knapsack, Longest Common Subsequence finding problem, LCS and LIS/LDS variations, Matrix Chain Multiplication. Red black Tree and Binomial Heaps, Stassen's algorithm. **Graph**: Strongly Connected Components, Euler Path, Articulation Point, Bridge, Bi-connected Components, Dijkstra's Shortest Path Algorithm, Bellman –Ford algorithm and negative cycle detection, Floyd-Warshall all pair shortest path algorithm, shortest path in Directed Acyclic Graph. **Minimum Spanning Tree (MST):** Prim's algorithm and Kruskal's algorithm. **Network Flow:** Flow Networks, Max-Flow Min-Cut Theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum Bipartite Matching, minimum path cover, minimum edge cover. **Matrix Operation:** Matrix Chain Multiplication. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. Traveling salesman problem. **Geometric algorithm:** Properties of Line-Segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. **RSA public key generation,** NP Completeness, NP hard and NP complete problems. **String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree and Suffix Array. Basic combinatorics, Probability and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT and FFT.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement all the listed algorithms
CLO 2	Construct bug free and efficient codes for the algorithms
CLO 3	Construct efficient solution of complex problems using suitable algorithms
CLO 4	Analyze the computational and memory complexities of algorithms
CLO 5	Prove the correctness of algorithms
CLO 6	Explain how and why the algorithms work
CLO 7	Apply the algorithms to solve real life problems
CLO 8	*Decide when to use which algorithm*
CLO 9	Synthesize algorithms to design complex solutions

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PLO of corresponding department.

Textbook

1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. 102 Combinatorial Problems by Titu Andreescu & Zuming Feng
3. Problem-Solving Methods in Combinatorics by Pablo Soberón
4. Algorithms by Robert Sedgewick and Kevin Wayne

Course Title: Computer Architecture		**Credits:** 3.0
Course Code: 0714 xx19*	**Contact Hours:** 3 hours/week	

Rationale:

This course is designed to provide a strong foundation for students to understand the modern areas of computer architecture. It will describe a broad range of architectural designs highlighting the design decisions and how these design decisions impact on system performance. The students will be able to apply these insights and principles to future computer designs.

Objectives:

- To make the students recognize the fundamental technologies and performance evaluation of different computer systems.
- To help them to know the instruction set architecture of a system and variations of ISA in different systems.
- To describe how computer performs arithmetic operations.
- To provide ideas about internal architecture of a processor along with parallel computing.
- To identify the underlying technologies on different levels of memory hierarchy and their management in a system.
- To accumulate basic ideas about fundamental technologies on multicore and multiprocessing system and their application.

Course Contents:

Introduction to Computer Architecture: Overview and history; Cost factor; Performance metrics, Fundamental blocks of computer. **Instruction set architecture: C**lassifying instruction set architectures, Registers, Addressing Modes, RISC versus CISC, x86 Architecture, ARM Architecture. **Memory Hierarchy:**Hierarchical Organization, Cache memory; Basic cache structure and design; Fully associative, direct, and set associative mapping; Analyzing cache effectiveness; Replacement policies; Writing to a cache; Multiple caches; Upgrading a cache; Main Memory; Virtual memory and machine, Paging, Replacement strategies. **Data Representation:** Data type representation, signed number, fixed point, floating point, character. **Processor and Pipelining:**Datapath, pipelined Datapath,Pipelining basics, types, stalling and forwarding, Throughput and Speedup of Pipelining, Pipelining hazards. **Parallelism:** Instruction level parallelism, introduction, challenges and limitations, Scalar and superscalar pipelining, branch prediction, increase uniprocessor throughput. **Multiprocessors and Multi-core Computers:** SISD, SIMD, and MIMD architectures; Centralized and distributed shared memory- architectures; Multi-core Processor architecture. **Input/output Devices:** Performance measure, Types of I/O device, Buses and interface to CPU, RAID.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Identify the fundamental technologiesincorporated in computer architectures.
CO 2	Elevate the memory management technologies.
CO 3	Implement pipelining mechanism and parallel computing in to the processor.
CO 4	Improve I/O performance.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PLO of the corresponding department

Textbook

1. Computer Architecture and Organization by John P.Hayes.
2. Computer Organization and Design: The hardware / software interface by David A.Patterson and John L.Hennessy.

Ordinance for the Graduate Program at SUST

Formation of Graduate Study Committee (GSC) will be the prerequisite to start a Graduate program in any Discipline. The GSC will be headed by the Head of the Discipline/ Institute consisting of all professors/Associate Professors of the discipline concerned with a minimum number of 3 professors/Associate Professors. When Professors and Associate Professors are not available in the discipline, the required number of Professors, Associate Professors will be included from the relevant Discipline/Institute by the proposal of the Board of Advanced Studies (BAS) & the Academic council.

1. Introduction

1.1 The graduate program consists of Masters (General), Masters (Thesis), M.Sc. (Engineering), Masters of Philosophy (M.Phil.) and Ph.D. degrees.
1.2 A graduate program may also be offered by a discipline in some specified field in collaboration with other disciplines.
1.3 Any student with (i) 4 year Bachelors degree (ii) 3-year Bachelor and 1-year Masters Degree or (iii) 5-year Bachelor of Architecture degree from a recognized university is eligible to get admitted into the graduate program at SUST.
1.4 Notification for the admission process will be published every year.
1.5 After admission every student will be assigned to a student advisor/supervisor from among the teachers of his/her discipline to guide him/her throughout the academic program.

2. Qualification

2.1 Masters and M.Phil.

2.1.1 Any student with a Bachelors degree from SUST is eligible for admission to the Masters (General) Program.

2.1.2 Any student with a CGPA of 3.25 or more from SUST is eligible for admission to the Masters (Thesis), M.Phil. (Engineering) or M.Phil. Program.
2.1.3 Four-year Graduates from other recognized universities and institutions with a CGPA of 3.25 or more can apply for admission to the Masters (Thesis), M.Phil. (Engineering) or M.Phil. Program. A candidate who passed under course system and seeks admission to M.Phil program has to have First class in Masters or 50% marks in Masters and at least 2nd division in all public examination.
2.1.4 Any student registered for Masters (General) or Masters (Thesis) may transfer to the M.Phil. program, offered by the relevant discipline, if he/she can maintain a CGPA of 3.25 or more during the first two semesters.
2.1.5 The GSC of a discipline will decide if a student from a related discipline will be allowed to apply to the graduate program of that discipline. In these cases if necessary the GSC may ask the candidate to take extra undergraduate/graduate courses to ensure the basic foundation.

2.2 Ph.D.

2.2.1 Candidates with Masters (Thesis), M.Phil. or M.Sc. (Engineering) Degrees are eligible for application for Ph.D. and will be selected after a written and/or viva voce examination and the proper evaluation of academic records by the GSC. A candidate who passed under course system and seeks admission to Ph.D. program has to have First class in Masters or 50% marks in Masters and at least 2nd division in all public examination.
2.2.2 A Masters (Thesis) an M.Phil. or an M.Sc. (Engineering) student may be transferred to the Ph.D. program after the completion of first two semesters with a CGPA 3.25 and the recommendation of his/her supervisor certifying satisfactory progress of research work and with the approval of the GSC and BAS.

2.2.3 The following candidates are eligible for direct admission to Ph.D. if they have a CGPA of 3.25 or more at Bachelors and Masters Level and 3.00 or equivalent in all public examinations. (i) University teachers with two years teaching experience and one publication in standard academic journals. (ii) Teachers of colleges with three years of teaching experience and one publication in a standard academic journal (iii) Researchers of recognized research organizations with three years of research experience and at least three publications in standard academic journals. (iv) Candidates with an M.Phil degree.

3. Admission

3.1 Masters and M.Phil.

3.1.1 If a SUST graduate has the required qualifications he/she can be admitted to the Masters program (General, Thesis or Engineering) as per the recommendation of the GSC.
3.1.2 The candidates for Masters (Thesis and Engineering) and M.Phil. will be selected for admission after a written and/or viva voce examination conducted by the GSC. Full time teachers of SUST are not required to sit for the admission test. GSC will then recommend the candidates for admission to the academic council through the BAS. During the process of admission each candidate shall be assigned by the appropriate GSC and approved by BAS a supervisor from among the teachers of the relevant discipline/institute not below the rank of an associate professor or an assistant professor with a Ph.D. / M.Phil. / M.S.

3.2 Ph.D.

3.2.1 A candidate for admission to the Ph.D. degree program will apply in the prescribed form to the head of the discipline or the director of institute along with the recommendation from possible supervisor(s). The supervisor must be of the rank of professor or associate professor.
3.2.2 After approval from the GSC, the application will be forwarded to the BAS for the approvals of the supervisor and co-supervisors (if any). Each candidate shall have not more than two co-supervisors; one co-supervisor may be from outside SUST. After careful scrutiny of the research proposal BAS will send it to the Academic Council for final Approval.
3.2.3 If necessary a change of supervisor must also be approved by the BAS and the Academic Council.

4. Registration

4.1 Every selected candidate will be registered with the University and enrolled as a full time or if allowed, part time student with payment of prescribed fees and dues before the commencement of each semester.
4.2 A student has to register for at least 50% or maximum 150% credits of the courses at every semester in the prescribed syllabus. But for attaining degree in the last semester above mentioned restrictions will not be followed.
4.3 A candidate may be admitted or change his status into part time student with prior approval of the university and a written consent from the serving organization. A part time student may be assigned a minimum of 6 credit hours per semester.
4.4 A full time student must register for a minimum of 1(2.0 Credits) hours per semester. A full time student shall not be allowed to be employed as a part time employee in other organizations. However he/she may be employed as teaching/research assistant at the University. A Ph.D. candidate shall have to be a full time student for at least one year during his/her Ph.D. work.
4.5 The registration for the Ph.D. degree will remain valid for a period of four years, and can be renewed for a further period of two years.

5. Academic Regulations

5.1 Duration

5.1.1 The minimum duration for the Masters, M.Sc. (Engineering), M.Phil. and Ph.D. degrees will be as followed:

Degree	Duration of Completion	Required Credits
Master of Engg. in CSE (Coursework)	2 Semesters	40
Master of Engg. in CSE (Mixed Mode)	3 Semesters	40
Master of Science in CSE (Research)	4 Semesters	40
Doctor of Philosophy in CSE	6 semesters	72

5.1.2 Minimum duration of M.Phil will be 4 Semesters for students who completed 3 years Bachelors and 1 year Masters degree. Minimum duration of M.Phil will be 2 semesters for students who completed 4 years Bachelors and 1 year Masters degree.

5.2 Credit Requirement

5.2.1 For the graduate program a full time student has to register for at least 1(2.0 Credits) each semester. For course work 1 credit means one hour of contact hour per week and for research or project work 1 credit hour means at least three hours per week. A student will be allowed to take theoretical course and research work simultaneously. Once the course requirement is completed, for the research work a graduate student has to register for "independent study" as credit/no-credit basis to fulfill the 1(2.0 Credits) per semester requirement.

5.3 Course Requirement

5.3.1 Syllabus committee for the graduate program will be comprised of the GSC members and two external members from other universities nominated by the Dean.

5.3.2 Every year the syllabus committee will design the graduate level courses for the respective disciplines and recommend the courses for approval of the Academic Council through the School and BAS. GSC can review the curriculum from time to time and recommend any change to the syllabus committee as may be considered necessary.

Masters and M.Phil.

5.3.3 Every Masters (general, thesis and engineering) and M.Phil. student has to complete at least 16 hours of theory course work during the first two semesters. GSC will propose the required courses to the students with consultation of respective supervisors. The course work for M.Phil Program may be reduced and relaxed according to the recommendation of GSC. In that case the duration may be reduced up to 1 year.

Ph.D.

5.3.4 The GSC may suggest courses, if felt necessary, for the Ph.D. students.

5.4 Research Work Requirement

5.4.1 Research work for thesis shall be carried out under the supervision of the supervisor. Co-supervisors from within or outside the discipline / Institute may be appointed, if necessary. The topic of research proposal shall be approved by the BAS after the completion of the required course credits within six months/one year for M.Phil. / Ph.D. on the recommendation of the Head of the Discipline/Institute. A Ph.D. student must submit a progress report of his work to the supervisor(s) at the end of the every semester who will present it to BAS.

5.4.2 The Ph.D. student will give at least one public seminar talk conducted by GSC at the Discipline / Institute every year on a topic of his own field of research.
5.4.3 The research work must be carried out in this University or at a place approved by the supervisor in consultation with the GSC.

6. Conduct of Examinations

6.1 Course Examination

6.1.1 The examination committee will conduct the course examinations as per the examination ordinance of graduate program.

6.2 Thesis Submission

6.2.1 The title of the thesis has to be approved by the BAS on the recommendation of the Head of the Discipline / Institute. For Masters/M.Phil. it has to be done at least three months and for Ph.D. it has to be done at least six months before submitting.
6.2.2 Every student shall submit to the supervisor required number of type written copies of his thesis in the approved forMATon or before a date to be fixed by the Head of the Discipline/ Institute in the consultation with the supervisor concerned.
6.2.3 The student shall declare that the research work was done by him/her and has not submitted elsewhere for other purpose (except for publication).
6.2.4 The thesis should demonstrate an evidence of satisfactory knowledge in the field of research undertaken by the student.

6.3 Masters Thesis Examination

6.3.1 There is no thesis requirement for Masters (General). The project (if any) and the thesis for Masters (Thesis) and will be evaluated as per the examination ordinance of graduate program.

6.4 M.Phil. / M.Sc. (Engineering) Thesis Examination

Thesis Evaluation
6.4.1 The academic council will, on the basis of the suggestion of the GSC and recommendation of the BAS, appoint for every thesis an examination committee consisting of two examiners of whom at least one shall be from outside this University.
6.4.2 The examiners of thesis will either accept it or reject it for the degree and then individually and separately submit one copy of their reports in sealed covers to the controller of examination and another copy to the GSC Chairman. The majority decision will be considered as the final result.
6.4.3 If a thesis is adjudged inadequate for the award of the degree, the candidate will be allowed to resubmit his thesis within six months. If the candidate fails to resubmit or the thesis is adjudged inadequate again the examiners may recommend Masters (general) degree and the controller of examination will place such recommendation before the BAS for the approval of academic council.

Oral Examination and Open Presentation
6.4.4 The GSC in consultation with the supervisor shall suggest, to the Vice Chancellor through BAS, a committee of three members for oral examination consisting of: (i) Convener: Thesis supervisor (ii) A Professor in relevant field from outside the University (iii) One of the thesis examiners.
6.4.5 If any examiner is unable to accept the appointment or has to relinquish his appointment before/ during the examination, the Vice-Chancellor shall appoint another examiner in his place as per the recommendation of GSC.
6.4.6 After the oral examination the convener will send a consolidated report to the controller of examinations stating clearly whether the award of the degree is recommended, who

will in turn place it to BAS for the approval of the Academic Council.

6.4.7 In case a candidate performs unsatisfactorily in oral examination even though the thesis is adjudged adequate the examiners may recommend to the Academic Council that the candidate may be permitted to appear at another oral examination within six months from the first oral examination. No candidate shall be allowed to appear at the oral examination of the same thesis for more than two times.

Recommendation for Degree

6.4.8 After completion for the viva-voce examination, the convener of the viva examination committee will send a consolidated report, stating clearly whether the award of the degree is recommended, to the Controller who will in turn place it to BAS for the approval of the academic council.

6.5 Ph.D. Thesis Examination

Thesis Evaluation

6.5.1 The academic council will, on the basis of the suggestion of the GSC and recommendation of the BAS, appoint for every thesis an examination committee consisting of three examiners of whom one shall be the supervisor and the other two from outside this University and at least one from a university from abroad

6.5.2 One of the three examiners will be appointed by the academic council as the convener of the examination committee.

6.5.3 The examiner of thesis will individually and separately submit one copy of their reports in sealed covers to the controller of examination and another copy to the convener. Every examiner will have to explicitly state whether the award of the Ph.D. degree is recommended or not. The

recommendations of all the three examiners must be explicit, unambiguous and unanimous for the award of the degree.

6.5.4 If a thesis is adjudged inadequate for the award of the Ph.D. degree, the candidate will be allowed to resubmit his thesis after six months with proper modification. If the candidate fails to resubmit or the thesis is adjudged inadequate again the examiners may recommend the award of M.Phil. or M. S. degree and the controller of examination will place such recommendation before the BAS for the approval of academic council.

Oral Examination and Open Presentation

6.5.5 On receipt of the unanimous opinions of the examiners, the convener shall fix a date and a venue and suggest, to the Vice Chancellor through BAS, a committee of three members for oral examination consisting of the convener, supervisor/co-supervisor and a thesis examiner. At least one of them has to be from outside the university.

6.5.6 If any examiner is unable to accept the appointment or has to relinquish his appointment before/during the examination, the Vice-Chancellor shall appoint another examiner in his place as per the recommendation of GSC.

6.5.7 In case a candidate is unable to satisfy the viva voice Board even though the thesis is adjudged adequate the Board may recommend to the Academic Council that the candidate may be permitted to appear at another oral examination after a lapse of six months from the first oral examination. No candidate shall be allowed to appear at the oral examination of the same thesis for more than two times.

Recommendation for Degree

6.5.8 After completion of the viva voce examination, the convener will send a consolidated report to the controller of examinations stating clearly whether the award of the degree is recommended, who will in turn place it to BAS for the approval of the Academic Council.

7. Award of the Degree

7.1 Masters

7.1.1 Students will be awarded his/her degree as per the recommendation of GSC chairman after the completion of his required credits.

7.2 M.Phil. and Ph.D.

7.2.1 The vice chancellor shall place the reports of the Oral Examination committee for consideration of the academic council which shall recommend to the Syndicate for the award of the degree.

7.2.2 A hard copy of the thesis accepted by the academic council incorporating any correction and changes suggested by the examination committee shall be preserved in the central library of the university and the corresponding electronic version shall be preserved in the archive.

8. Academic Fee

8.1 To be decided by the Academic Council and the Syndicate.

Ref.:The clause 4.2 of this Ordinance was approved in the 119th Academic Council.

Examination Ordinance for the Graduate Program

University authorities will administer and publish the results of Masters, M.Phil. and Ph.D. degree examinations under the graduate program. The graduate program will follow the same academic calendar of the undergraduate program for course delivery, the final examination and publication of results. The graduate courses are comprised of theory and lab courses and where applicable, the thesis for the research works. The evaluation of thesis is conducted as per the Ordinance for the Graduate Program at SUST. The theory and lab courses are conducted by the examination committee.

1. Examination Committee

1.1 The GSC of the Discipline/Institute will form the examination committee as per the rules of the University.
1.2 The examination committee will propose the examination schedule, prepare question papers, help the discipline conducting the examination, prepare results and will resolve the issues that may arise concerning the examination procedure.

2. Examination Dates and Routines

2.1 The examination routines will be designed by the respective disciplines and Head of the disciplines will notify them and send copies to the other relevant disciplines and to the office of the Controller of the Examinations.

3. Theory Courses

3.1 Distribution of Marks
A student will be continuously evaluated during the semester through tests, assignments, mid-semester examinations, viva etc. conducted by the course teachers, and it will contain 30% of total marks. The rest 70% marks will come from the final written examination at the end of that semester.

3.2 Class Performance
After the end of the classes, the course teachers will make three copies of mark-sheets showing the marks from class participation and assignment and mid semester examination. He/she will display one copy in the notice board, send one sealed copy to the chairman of the examination committee and another sealed copy to the controller of examination.

3.3 Question Setting and Moderation
3.3.1 The examination Committee will appoint two question setters for each course at least four weeks before the date of commencement of the examination and inform the Controller of examination. The controller of examination will send the necessary papers to the question setters and the examiners. If a question setter or examiner declines the responsibility, he/she will return all the papers and the examination committee will suggest an alternative question setter or examiner.
3.3.2 The chairman of the examination committee will receive all the manuscript of question papers; if no manuscript is received within the specified time the committee will suggest an alternative question setter.
3.3.3 After receiving all the question papers the examination committee will moderate the question papers. Moderation will not be invalid if any member be absent during moderation. For the disciplines of the school of Applied Sciences and Technology the questions will be divided in two groups in the question paper so that two examiners can evaluate the answer script simultaneously. The examination committee will be responsible for the preparation of the necessary editing and printing of the question papers.

3.4 Final Examination

3.4.1 The controller of examination will be responsible to print the blank answer scripts, mark sheets and other relevant forms and will make necessary arrangements, so that these are available during the conduct of examination in the examination hall in due time .

3.5 Evaluation of Answer Script

3.5.1 The answer scripts from the disciplines of Applied Science and technology will be evaluated by two examiners simultaneously, of whom one should preferably the course teacher. The answer scripts from the disciplines of other school of studies will be evaluated by two examiners separately, of whom one should preferably the course teacher. The examiners will examine the scripts thoroughly, mark the scripts properly and grade legibly within the specified time. The examiners will send a sealed copy of mark-sheet to the controller of examination and one sealed copy to the chairman of the examination committee.

3.5.2 The examination committee will assign members from the committee to scrutinize the answer scripts and if any discrepancy is found the committee will make the necessary arrangements to fix the problem and inform the controller of examination.

3.5.3 If the difference between marks given by two examiners be 20% or more than 20% GSC will recommend a third examiner for approval by the V.C and marks given by 3rd examiner & the marks of the first or 2nd examiner which ever is nearest to this will be considered for the average marks .

4. Lab Courses

4.1 Every lab course will be assigned to at least two course instructors and they will grade the students through continuous evaluation.

4.2 For the projects, Masters (Thesis), Industrial assignments, monographs etc. the supervisor will give an overall assessment which will count as 30% of the total marks. Evaluation of the report by two external examiners, who is not involved in supervision/co-supervision will count as another 30% of the marks. The remaining 40% will come from the presentation and viva voce conducted by the examination committee. During viva-voce examination the supervisor or co-supervisor, if present, will not participate in marking.

5. Publication of Result

5.1 Three original tabulation sheets will be prepared by the tabulators and checked by all the members and signed by the tabulators and members of the examination committee. The tabulation sheets will contain the grade point average obtained in the specific semester. The tabulation sheets will be sent to the Controller of Examinations for his signature and approval by the Vice-Chancellor.

5.2 The Controller of Examination shall keep up to date record of all the grades obtained by the student in individual Academic Record Card. Grades shall be announced by the Controller of Examination at the end of each semester.

Grade and grade points:

5.3 The letter grade and grade point will be awarded as follows:

Numerical Grade	Letter Grade	Grade Points
80% Or above	A+	4.00
75% to less than 80%	A	3.75
70% to less than 80%	A-	3.50
65% to less than 70%	B+	3.25
60% to less than 65%	B	3.00
55% to less than 60%	B-	2.75
50% to less than 55%	C+	2.50
45% to less than 50%	C	2.25
40% to less than 45%	C-	2.00
Less than 40%	F	0.00

6. Security and Ethics

6.1 Everyone involved in the process of examination has to guard the security of the question papers, examination grades and the final results. An examinee can never try to influence the examiners and any such attempt has to be brought to the controller of examination.

6.2 A student may never be asked a question so that he is hurt because of his religious or ethnic background.

6.3 If some one involved in the examination process has the following relatives as examinee he/she should immediately inform in to the authority: (a) Husband/wife, (b) Son/Daughter, (c) Brother/Sister, (d) Brother-in-Law/ Sister-in-Law (e) Son-in-Law/ Daughter-in-Law, (f) Nephew/ Niece, (g) Uncle/ Aunt, (h) First Cousins.

Department of Computer Science and Engineering
Graduate Program
Session 2024-2025

Department of Computer Science and Engineering (CSE) offers the following graduate programs:

i. Master's by Coursework
ii. Master's by Mixed Mode
iii. Master's by Research
iv. Doctoral by Research

The curriculum for the Graduate programs in CSE covers the requisite courses for the following degrees:

i. Master of Engg. in CSE (Coursework)
ii. Master of Engg. in CSE (Mixed Mode)
iii. Master of Science in CSE
iv. Doctor of Philosophy in CSE

A. Master's by Coursework

Credit requirement and duration of Master's by Coursework program are as follows:

Credit Requirement		Program Duration			
Coursework (Min.)	Total (Min.)	Semester (Min.)	Year (Min.)	Semester (Max.)	Year (Max.)
40	40	2.0	1.0	04	2.0

The course structure for this program is given below.

Semester I

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.
			Theory	Lab	
CSE	0619 5193	Deep Learning and Neural Network	4	0	4.0
CSE	0714 5175	Advanced Digital Image Processing	4	0	4.0
GED	0611 5159	Research Methodology for Engineers	4	0	4.0
CSE	0*** 51**	Option I	4	0	4.0
CSE	0*** 51**	Option II	4	0	4.0
Total			20		20.0

Semester II

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.
			Theory	Lab	
CSE	0619 5275	Digital Speech Signal Processing	4	0	4.0
CSE	0613 5243	Natural Language Processing	4	0	4.0
CSE	0613 5241	Software Testing and Quality Assurance	4	0	4.0
CSE	0*** 52**	Option III	4	0	4.0
CSE	0*** 52**	Option IV	4	0	4.0
Total			20		20.0
Grand Total					40.0

B. Master's by Mixed Mode

Credit requirements and duration of Master's by Mixed Mode program are as follows.

The course structure for this program is given below.

Credit Requirement				Program Duration			
Coursework (Min.)	Dissertation (Min.)	Dissertation (Max.)	Total (Min.)	Semester (Min.)	Year (Min.)	Semester (Max.)	Year (Max.)
18	12	16	40	3.0	1.5	6.0	3.0

Semester I

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.
			Theory	Lab	
CSE	0619 5193	Deep Learning and Neural Network	4		4.0
CSE	0714 5175	Advanced Digital Image Processing	4		4.0
STA	0611 5159	Research Methodology for Engineers	4		4.0
CSE	0*** 51**	Option I	4		4.0
CSE	0610 5350	Thesis	0	12	4.0
Total			28		20.0

Semester II

Offering Dept.	Course Code	Course Title	Hours/Week		Cr.
			Theory	Lab	
CSE	0619 5275	Digital Speech Signal Processing	4		4.0
CSE	0*** 52**	Option II	4		4.0
CSE	0610 5350	Thesis (continued)	0	12	4.0
Total			20		12.0
Grand Total					32.0

Semester III

Offering Dept	Course Code	Course Title	Hours/Week		Credits
			Theory	Lab	
CSE	0610 5360	Thesis (continued)	0	24	8.0
	Total		24		8.0
	Grand Total				40.0

C. Master's by Research

Credit requirements and duration of Master's by Research program are as follows.

Credit Requirement			Program Duration			
Coursework (Min.)	Dissertation (Min.)	Total (Min.)	Semester (Min.)	Year (Min.)	Semester (Max.)	Year (Max.)
Non-credit (if offered)	48	48	4.0	2.0	6.0	3.0

The course structure for this program is given below.

Semester I

Offering Dept.	Course Code	Course Title	Hours/Week		Credits
			Theory	Lab	
CSE	0610 5460	Independent Study	0	36	12.00
	Total		36		12.00

Semester II

Offering Dept.	Course Code	Course Title	Hours/Week		Credits
			Theory	Lab	
CSE	0610 5460	Independent Study (continued)	0	36	12.00
Total			36		12.00
Grand Total			72		24.00

Semester III

Offering Dept.	Course Code	Course Title	Hours/Week		Credits
			Theory	Lab	
CSE	0610 5460	Independent Study (continued)	0	36	12.00
Total			36		12.00
Grand Total			108		36.00

Semester IV

Offering Dept.	Course Code	Course Title	Hours/Week		Credits
			Theory	Lab	
CSE	0610 5460	Independent Study (continued)	0	36	12.00
Total			36		12.00
Grand Total			144		48.00

Optional Courses

Course Code	Course Title	Hours/Week Theory and Lab
0611 5159/5259	Research Methodology for Engineering	4
0612 5161/5261	Applied Cryptography and Network Security	4
0613 5133/5233	Quantum Mechanics and Quantum Computing	4
0613 5137/5237	Algorithm and Complexity Analysis	4
0613 5141/5241	Software Testing and Quality Assurance	4
0613 5163/5263	Big Data Analytics	4
0613 5165/5265	Graph Theory	4

0613 5167/5267	Distributed Computing System	4	4.0
0613 5191/5291	Data Mining and Information Retrieval	4	4.0
0613 5195/5295	Computer Vision	4	4.0
0613 5197/5297	Advanced Bioinformatics	4	4.0
0714 5173/5273	IoT and Embedded Systems	4	4.0
0611/0612/0613/0619 5187	Contemporary Course on Computer Science and Engineering I	4	4.0
0611/0612/0613/0619 5287	Contemporary Course on Computer Science and Engineering II	4	4.0

D. Doctoral by Research
Credit requirements and duration of Doctoral by Research program are as follows.

The course structure for this program is given below.

Detailed Syllabus for Master's Program

Course Title: **Deep Learning and Neural Networks**		Credits: 4.0
Course Code: 0619 5193	Contact Hours: 4 hours/week	

Rationale:
Most of the advances in artificial intelligence in the last few years have come from deep learning. Neural networks are behind some of the most disruptive and awe-inspiring applications, including autonomous cars, chatbots, image recognition, and robotics. And, with the latest developments in generative AI, the field of deep learning is under more scrutiny than ever before.

Objectives:
make the students understand the theoretical basis underlying neural networks and deep learning.
develop skills for implementation of neural components as well as applying deep learning on real-world datasets using modern deep learning packages.

Course Content:
Introduction: The Road to Deep Learning, The Essence of Deep Learning. Preliminaries: Data Manipulation and Preprocessing, Linear Algebra, Calculus, Automatic Differentiation, Probability and Statistics. Linear Neural Networks for Regression: Linear Regression, Object-Oriented Design, Synthetic Regression Data, Linear Regression Implementation, Generalization, Weight Decay. Linear Neural Networks for Classification: Softmax Regression, The Base Classification Model, Softmax Regression Implementation, Generalization in Classification, Environment and Distribution Shift. Multilayer Perceptrons: Implementation of Multilayer Perceptrons, Forward Propagation, Backward Propagation, and Computational Graphs, Numerical Stability and Initialization, Generalization in Deep Learning, Dropout. Convolutional Neural Networks: Fully Connected Layers to Convolutions, Convolutions for Images, Padding and Stride, Multiple Input and Multiple Output Channels, Pooling, Convolutional Neural Networks (LeNet), Deep Convolutional Neural Networks (AlexNet), Networks Using Blocks (VGG), Network in Network (NiN), Multi-Branch Networks (GoogLeNet), Batch Normalization, Residual Networks (ResNet) and ResNeXt, Densely Connected Networks (DenseNet), Designing Convolution Network Architectures. Recurrent Neural Networks: Working with Sequences, Converting Raw Text into Sequence Data, Language Models, Recurrent Neural Network Implementation, Backpropagation Through Time, Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Deep Recurrent Neural Networks, Bidirectional Recurrent Neural Networks, The Encoder-Decoder Architecture, Beam Search. Attention Mechanisms and Transformer: Attention Pooling by Similarity, Attention Scoring Functions, The Bahdanau Attention Mechanism, Multi-Head Attention, Self-Attention and Positional Encoding, The Transformer Architecture, Large-Scale Pretraining with Transformers. Optimization Algorithms: Optimization and Deep Learning, Convexity, Gradient Descent, Stochastic Gradient Descent, Minibatch Stochastic Gradient Descent, Momentum, Adagrad, RMSProp, Adadelta, Adam, Learning Rate Scheduling. Hyperparameter Optimization: Asynchronous Random Search, Multi-Fidelity Hyperparameter Optimization, Asynchronous Successive Halving. Generative Adversarial Networks: Deep Convolutional Generative Adversarial Networks. Applications: Computer Vision, Natural Language Processing, Recommender Systems.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	explain the basic principles behind neural networks and deep learning
CO 2	compare modeling aspects of various neural network architectures
CO 3	implement simple neural network algorithms
CO 4	apply and evaluate deep learning on real datasets

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO3	PO 4	PO5	PO6	PO7	PO 8	PO9	PO1 0	PO1 1	PO1 2
CO1	3					2						
CO2	3	3		3								
CO3			3		3		2		2			
CO4							2		3		2	

Textbook

Dive into Deep Learning, By Aston Zhang, Zachary C. Lipton, Muli, and Alexander J. Smola

Deep Learning with Python, By Francois Chollet

Course Title: **Digital Speech Signal Processing**		Credits: 4.0
Course Code: 0619 5175	Contact Hours: 4 hours/week	

Rationale:

This course is designed to equip students to:

a. familiarize with modeling the vocal tract as a digital, linear, time-invariant system.

b. convey details of a range of commonly used speech feature extraction techniques.

c.familiarize yourself with the practical aspects of speech processing, including robustness and applications of speech processing, including text-to-speech systems, speech enhancement, speaker recognition, and speech recognition.

d. gives you practical experience with implementing several components of speech processing systems.

Objectives:

Express the speech signal in terms of its time domain and frequency domain representations and the differentways in which it can be modeled;

Derive expressions for simple features used in speech classification applications.

Explain the operation of example algorithms covered in lectures and discuss the effects of varying parametervalues within these.

Synthesize block diagrams for speech applications, explain the purpose of the various blocks, and describe indetail algorithms that could be used to implement them.

Implement components of speech processing systems, including speech recognition and speaker recognition, in Python/ MATLAB.

Deduce the behaviour of previously unseen speech processing systems and hypothesize about their merits.

Course Content:

Introduction to Digital Speech Processing; Review of DSP Fundamentals; Acoustic Theory of Speech Production; Speech Perception--Auditory Models, Sound Perception Models, MOS Methods; Sound Propagation in the Vocal Tract; Time Domain Methods in Speech Processing; Short Time Fourier Analysis Methods--Filter Bank Summation and Overlap Add; Speech Representations Based on STFT Analysis-Synthesis Methods;

Homomorphic Speech Processing; Linear Predictive Coding (LPC) Methods; LPC--Frequency Domain Interpretations, Methods for Synthesis and Vocoding; Speech Waveform Coding;Speech analysis: pitch and period extraction, formant estimation, voiced - unvoiced decision, Linear prediction, Inverse filtering; Auditory modelling, auditory masking; Speech enhancement; Clustering, Gaussian mixture modelling, hidden Markov modelling, Implementation of speech and speaker recognition systems, Text-to-speech systems.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Explain Digital Signal Processing Fundamentals and Speech Representation Algorithms
CO 2	Extract Features from Speech Signals
CO 3	Develop speaker and speech recognition systems using machine learning approaches and Speech Synthesis and Text-to-Speech Conversion
CO 4	Assess and discuss ethical considerations related to speech signal processing, including issues related to privacy, bias, and security in speech-based applications.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												3
CO2		3										
CO3											3	
CO4			3	3								

Textbook

Quatieri, T. F. (2002). Discrete-Time Speech Signal Processing, Prentice-Hall, New Jersey.
Rabiner, L. R., and Juang, B.-H. (1993). Fundamentals of Speech Recognition, Prentice-Hall, New Jersey.

Course Title: **Advanced Digital Image Processing**		Credits: 4.0
Course Code: 0613 5231	Contact Hours: 4 hours/week	

Rationale:
This course provides an introduction to basic concepts, methodologies, and algorithms of digital image processing for image analysis and information retrieval.

Objectives:
To introduce the concepts of image processing and basic analytical methods to be used in image processing.
To familiarize students with image enhancement and restoration techniques.
To explain different image compression techniques.
To introduce segmentation and morphological processing techniques

Course Content:
Digital image fundamentals: Fundamentals of Image formation, components of image processing system, image sampling and quantization.
Image enhancement in the spatial domain: Basic gray-level transformation, histogram processing, arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters.

Image enhancement in the frequency domain: Introduction to Fourier Transform, Smoothing and Sharpening frequency domain filters, Ideal, Butterworth and Gaussian filters.
Image restoration: A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the image enhance in frequency domain.
Image Compression: Need of image compression, image compression models, error-free compression, lossy predictive coding, image compression standards.
Morphological Image Processing: Preliminaries, dilation, erosion, open and closing, basic morphologic algorithms, The Hit-or-Miss Transformation
Image Segmentation: Detection of discontinuous, edge linking and boundary detection, thresholding, Hough Transform Line Detection and Linking, region–based segmentation.
Object Recognition: Patterns and patterns classes, matching, classifiers.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO1	Explain the fundamentals of digital image and its processing as well as object detection and recognition technique
CO2	Perform image enhancement techniques in spatial and frequency domain
CO3	Elucidate the mathematical modeling of image restoration and compression
CO4	Apply the concept of image segmentation

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			2		3							
CO3			2		1							
CO4				3								

Textbook
Digital Image by R. C. Gonzalez and R.E. Woods (Pearson Prentice Hall)
Digital Image Processing Using MATLAB (Third Edition) by Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins
Fundamentals of Digital Image Processing by Anil Jain K.

Course Title: **Natural Language Processing**	Credits: 4.0
Course Code: 0613 5243	Contact Hours: 4 hours/week

Rationale:
As momentum for machine learning and artificial intelligence accelerates, natural language processing (NLP) plays a more prominent role in bridging computer and human communication. This course is intended as a theoretical and methodological introduction to the most widely used and effective current techniques, strategies and toolkits for speech and language processing.

Objectives:

- teach students the leading trends and systems in natural language processing.
- make them understand the concepts of morphology, syntax, semantics and pragmatics of the language and that they are able to give the appropriate examples that will illustrate the above mentioned concepts.

- enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

Course Content:
Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms, The State of the Art, Brief History. Lexical Computation: Regular Expressions, Words, Corpora, Text Normalization, Minimum Edit Distance. N-gram Language Models: N-grams, Evaluating Language Models, Smoothing. Naïve Bayes, Text Classification, and Sentiment: Naïve Bayes Classifiers, Text Classification, Sentiment Analysis, Test Sets and Cross-Validation, Evaluation: Precision, Recall, F-measure. Logistic Regression: Classification with Logistic Regression, Gradient Descent, Regularization. Semantics and Embeddings: Words and vectors, TF-IDF, PMI, Word2Vec, Embeddings. Neural Language Models: Transformers as Language Models, Pretraining Large Language Models, Language Models for Zero Learning. Sequence Labeling: Part-of-Speech Tagging, Named Entity Tagging, Conditional Random Fields. Parsing: Context-Free Grammars, Treebanks, CKY Parsing, Neural Constituency Parsing, Dependency Parsing. Computational Semantics: Logical Representations of Sentence Meaning, Relation and Event Extraction, Time and Temporal Reasoning, Word Senses and WordNet, Semantic Role Labeling, Lexicons, Coreference Resolution, Discourse Coherence. Phonetics: Speech Sounds and Phonetic Transcription, Articulatory Phonetics, Prosody, Acoustic Phonetics and Signals. Applications: Machine Translation, Question Answering and Information Retrieval, Chatbots and Dialogue Systems, Automatic Speech Recognition and Text-to-Speech.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	gain a foundational understanding in natural language processing methods and strategies.
CO 2	evaluate the strengths and weaknesses of various NLP technologies and frameworks as they gain practical experience in the NLP toolkits available.
CO 3	design solutions to real-life problems using NLP-based analytic techniques.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO9	PO10	PO11	PO12
CO1	3					2						
CO2	3	3		3	3							
CO3			3				2		2		3	3

Textbook
Speech and Language Processing (3rd Ed.), By Daniel Jurafsky and James H. Martin
Natural Language Processing with Python, By Steven Bird, Ewan Klein, Edward Loper

OPTIONAL COURSES

Course Title: **Research Methodology for Engineering**		Credits: 4.0
Course Code: 0611 5159/5259	Contact Hours: 4 hours/week	

Rationale:

The main aim of this course is to provide the students with the knowledge and skills required to conduct and document engineering research. The course allows the students to engage in different research activities such as research planning, literature review, simulation, experimentation, data analysis and reporting (written and oral) using a selected mechatronics research topic.

Objectives:

- Identify and formulate research problems
- Search for valuable information
- Evaluate and summarize journal papers
- Simulate mechatronics-related models
- Conduct engineering experiments
- Analyse results among different methods
- Write research proposals
- Write research paper

Course Content:

Introduction to Research Methodology, Research types and applications,, Research management, Research phases, Research problem formulation, Academic honesty, Literature Review, How to read journal papers, Literature review process, Information sources, Synthesizing information, Writing the literature review, Referencing, Thesis Proposals, Thesis Proposal main sections, How to write Thesis Proposals, Engineering Experiments, Experiment set-up (Lab Work), Running experiments (Lab Work), Data collection and analysis, Writing Research Papers , How to write research paper, Technical writing , Where to submit How to submit.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Recognize the ethical principles of conducting applied research. Identify various sources of information and formulate research problem.
CO 2	Carry out literature searches and ability to critically evaluate literature. Design/conduct experiments, devise appropriate measurements, analyse data and form conclusions.
CO 3	Undertake and manage a research projects. Documenting all aspects of the development of an engineering project.
CO 4	Apply project management skills to research activities. Communicate effectively in written and oral ways.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			3									
CO3				3				2				
CO4												3

Textbook

Herman Tang, Engineering Research: Design, Methods, and Publication, ISBN: 978-1-119-62448-6 December 2020

Course Title: **Applied Cryptography and Network Security**		Credits: 4.0
Course Code: 0612 5161/5261	Contact Hours: 4 hours/week	

Rationale:
This course is an introduction to the basic concepts and mechanisms of applied cryptography and data security. It will cover both cryptographic primitives (symmetric encryption, public encryption, MACs, Digital Signatures, Authenticated Encryption, etc.) to cope with the data confidentiality and data integrity. It also emphasizes on how to apply and implement cryptography in practice. (The content and syllabus are subject to adjustment during the semester.)

Objectives:

- Learn the main areas of Modern Cryptography, including their main problem statements and the rigorous mathematical approaches used to formalize them
- Learn and describe how various cryptographic algorithms and protocols work, pointing out the main techniques used in them, and proving/disproving most basic properties, such as correctness of decryption, digital signatures, authentication tags, and key agreement
- Evaluate functionality, security and performance properties of cryptography methods used as components of complex security solutions
- Analyze the impact of errors or different designs of cryptography algorithms and protocols
- Describe the applications of cryptography algorithms and protocols to real-life problems and many implementation issues in developing these solutions.
- Course Content:
- Cryptography; symmetric and public-key, cryptographic protocols, key distribution, authentication, digital signatures. Network security; access control, services and mechanisms for authentication, network security protocols, application layer security.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	To get a basic understanding of principles and practice of cryptography and network security
CO 2	Use of cryptographic algorithms in network protocols and network applications.
CO 3	To be able to evaluate security of systems with respect to appropriate use of security services and mechanisms;
CO 4	To be able to design basic security architectures through selection and integration of relevant security components.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			3									
CO3				3								
CO4					3							

Textbook
W. Stallings, Cryptography and Network Security: Principles and Practice, 2nd edition, Prentice Hall
N. Ferguson, B. Schneier and T. Kohno, Cryptography Engineering: Design, Principles and Practical Applications, Wiley Publishing, Inc., 2010

Course Title: **Quantum Mechanics and Quantum Computing**		Credits: 4.0
Course Code: 0613 5133/5233	Contact Hours: 4 hours/week	

Rationale:

This course is designed to provide a comprehensive understanding of the core principles of quantum information processing, encompassing areas such as quantum computation, quantum cryptography, and quantum information theory. Over the past two decades, significant advancements have been made, leading to the discovery of algorithms that directly leverage the principles of quantum mechanics to accelerate specific computations and ensure secure communication. Among these developments are rapid quantum algorithms capable of factoring large integers and computing discrete logarithms, which, if implemented, could compromise the security of prevalent encryption methods currently in use. This realization has catalyzed substantial and ongoing efforts towards constructing quantum computers, with several companies having introduced operational albeit somewhat limited quantum computing devices. Additionally, quantum computation holds promise for simulating intricate quantum systems like extensive molecules. Throughout this course, we will delve into the foundational aspects of quantum information processing, covering both theoretical principles and practical applications.

Objectives:

- Help the students to know the functioning method of quantum computer
- Help them conceptualize the basic quantum gates and their working techniques
- To facilitate necessary knowledge about quantum algorithms
- To facilitate necessary knowledge about implementing quantum circuits for different algorithms
- Learn about quantum cryptography

Course Content:

Fundamentals of quantum information processing include quantum computation, cryptography, and information theory. Topics include the quantum circuit model, alternative models, qubits, unitary operators, measurement, entanglement, quantum algorithms for factoring and search, quantum cryptographic key distribution, simulation of physical systems, error-correction and fault-tolerance, information capacity of quantum channels, the complexity of quantum computation, near-term implementations, quantum supremacy, and quantum advantage.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CO 1	Analyze the behavior of basic quantum algorithms
CO 2	Implement simple quantum algorithms and information channels in the quantum circuit model
CO 3	Simulate a simple quantum error-correcting code
CO 4	Analyze the cryptographic protocols

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3										
CO2			3		2							
CO3		2		2								
CO4		2		1								

Textbook
Quantum Computing for Computer Scientists by Mirco A. Mannucci and Noson S. Yanofsky
Quantum Computing for Everyone by Chris Bernhardt

Course Title**: Algorithm and Complexity Analysis**		Credits: 4.0
Course Code: 0613 5137/5237	Contact Hours: 4 hours/week	

Rationale:
The need for efficient algorithms arises in nearly every area of computer science. But the type of problem to be solved, the notion of what algorithms are "efficient,'' and even the model of computation can vary widely from area to area. In this course, many of the techniques that apply broadly in the design of efficient algorithms will be surveyed. Their application in a wide range of application domains and computational models will be studied also.

Objectives:
- To familiarize with the asymptotic performance of advanced algorithms
- To familiarize with rigorous correctness proofs for advanced algorithms
- To demonstrate familiarity with major advanced algorithms
- To facilitate with the necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Content:
Data Structures: More Advanced Solutions to Basic Data Structuring Problems: Fibonacci Heaps. Van Emde Boas Priority Queues. Dynamic Data Structures for Graph Connectivity/Reachability.
Bit Tricks: Word-level Parallelism. Transdichotomous Model. o(n \log n) Integer Sorting.
String Algorithms: Rabin-Karp Fingerprinting Algorithm. Suffix Trees.
Maximum Flows: Augmenting Paths and Push-Relabel Methods. Minimum Cost Flows. Bipartite Matching.
Linear Programming: Formulation of Problems as Linear Programs. Duality. Simplex, Interior Point, and Ellipsoid Algorithms.
Online Algorithms: Ski Rental. River Search Problem. Paging. The k-Server Problem. List Ordering and Move-to-Front.
Approximation Algorithms: One Way of Coping with NP-Hardness. Greedy Approximation Algorithms. Dynamic Programming and Weakly Polynomial-Time Algorithms. Linear Programming Relaxations. Randomized Rounding. Vertex Cover, Wiring, and TSP.
Fixed-Parameter Algorithms: Another Way of Coping with NP-Hardness. Parameterized Complexity. Kernelization. Vertex Cover. Connections to Approximation.
Parallel Algorithms: PRAM. Pointer Jumping and Parallel Prefix. Tree Contraction. Divide and Conquer. Randomized Symmetry Breaking. Maximal Independent Set.
External-Memory Algorithms: Accounting for the Cost of Accessing Data from Slow Memory. Sorting. B-trees. Buffer Trees. Cache-oblivious Algorithms for Matrix Multiplication and Binary Search.
Computational Geometry: Convex Hull. Line-segment Intersection. Sweep Lines. Voronoi Diagrams. Range Trees. Seidel's Low-dimensional LP Algorithm.
Streaming Algorithms: Sketching. Distinct and Frequent Elements

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Explain theoretically various advanced algorithms and improve their practical problem solving skills using them.
CO 2	Explain a wide range of advanced algorithm design techniques.
CO 3	Analyze advanced algorithms.
CO 4	Develop solution to real-world problems using advanced algorithms

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3										
CO2	3	3	3									
CO3		3		3								
CO4		3		3	3							

Textbook

Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
102 Combinatorial Problems by Titu Andreescu &Zuming Feng
Problem-Solving Methods in Combinatorics by Pablo Soberón
Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: **Software Testing and Quality Assurance**		Credits: 4.0
Course Code: 0613 5141/5241	Contact Hours: 4 hours/week	

Rationale:
This course presents a practical approach to software testing as a sub-discipline of software engineering. It introduces software quality concepts, standards, measurements, and practices that support the production of quality software. It offers a solid foundation in testing fundamentals including test case design, test management, and test measurement strategies, improving software test processes' effectiveness. Software quality and software testing concepts are presented from managerial, technical and process-oriented perspectives. Software quality assurance is explored for both traditional, plan-driven software development processes and newer, agile development methodologies.

Objectives:

- Upon completion of this course, learners should be able to: Explain the similarities and differences in software quality practices between plan-driven and agile teams.
- Integrate software quality assurance practices using Testing Maturity Model (TMM) levels for software development processes.
- Develop a comprehensive software quality and test plan.
- Analyze test cases to support multiple testing goals.
- Develop a test automation strategy by selecting two or more automated test tools.
- Evaluate quantitative methods for evaluating the effectiveness of software quality processes.
- Discuss ethics and social responsibility related to current incidents within the Software Engineering community

Course Content:
Definition and concept of software quality assurance (SQA); quality models; specification of quality requirements; product development & delivery issues; software development processes & maturity; software quality management process: total quality management, improvement cycle, SQA planning & management, organizing the SQA effort; software verification & validation; typical software development errors; Fagan inspections; software audit; software testing: testing objectives & testing fundamentals, testing theory, coverage criteria, equivalence class testing, value-based testing, decision table, syntax & state transition testing, statement & path testing, branch & condition testing, data flow testing, thread-based testing, integration & integration testing, system testing; testing in object-oriented systems; test tools & test automation; test management; problem reporting & corrective action.

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Explain the basic concepts of software testing and quality assurance.
CO 2	Integrate software quality assurance practices using Testing Maturity Model (TMM) levels for software development processes.
CO 3	Develop a comprehensive software quality and test plan and analyze test cases to support multiple testing goals.
CO 4	Develop a test automation strategy by selecting two or more automated test tools.
CO 5	Evaluate quantitative methods for evaluating the effectiveness of software quality processes.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			3									
CO3					3							
CO4						3						
CO5								2				

Textbook
Software Testing: Concepts and Operations. Ali Mili, Fairouz Tchier. ISBN: 978-1-118- 66287-8. 368 pages.

Course Title: **Big Data Analytics**		Credits: 4.0
Course Code: 0613 5163/5263	Contact Hours: 4 hours/week	

Rationale:
This course covers the knowledge of big data science. It serves as a graduate level course for graduate students. The focus will be big data computer system, storage, processing, analysis, visualization, and applications. State-of-the-art computational frameworks for big data will be introduced to students. Students will learn the essentials of big data analytics

Objectives:

- Understand the Big Data Platform and its Use cases
- Provide an overview of Apache Hadoop
- Provide HDFS Concepts and Interfacing with HDFS
- Understand Map Reduce Jobs
- Provide hands on Hodoop Eco System
- Apply analytics on Structured, Unstructured Data.
- Exposure to Data Analytics with R.

Course Content:

Introduction to Big Data: Defining Big Data, Delivering business benefit from Big Data; Storing Big Data: Analyzing your data characteristics; Overview of Big Data stores: Data models, Hadoop Distributed File System, HBase, Hive, Cassandra, Hypertable, Amazon S3, BigTable, DynamoDB, MongoDB, Redis, Riak, Neo4J; Selecting Big Data Stores: Choosing the correct data stores based on your data characteristics, Moving code to data, Implementing polyglot data store solutions, Aligning business goals to the appropriate data store; Processing Big Data: Integrating disparate data stores, Employing Hadoop MapReduce, The building blocks of Hadoop, MapReduce, Handling streaming data; Tools and Techniques to Analyze Big Data: Abstracting Hadoop MapReduce jobs with Pig, Performing ad hoc Big Data querying with Hive, Creating business value from extracted data; Developing a Big Data Strategy: Defining a Big Data strategy for your organization, Enabling analytic innovation, Implementing a Big Data Solution

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Identify Big Data and its Business Implications.
CO 2	Access and Process Data on Distributed File System and manage Job Execution in Hadoop Environment
CO 3	Analyze Infosphere BigInsights Big Data Recommendations.
CO 4	Develop Big Data Solutions using Hadoop Eco System

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			3									
CO3				3								
CO4					3							

Textbook

Tom White " Hadoop: The Definitive Guide" Third Edit on, O'reily Media, 2012.

Seema Acharya, Subhasini Chellappan, "Big Data Analytics" Wiley 2015.

Course Title: **Graph Theory**	Credits: 4.0
Course Code: 0613 5165/5265	Contact Hours: 4 hours/week

Rationale:

This advanced-level course focuses on delving into algorithmic graph theory, where participants will delve into its fundamental theorems, essential problems, and effective algorithms and tools. The primary objective is to systematically present crucial outcomes to facilitate the creation of graph algorithms. Mathematical characteristics of graphs will be leveraged to devise novel algorithms and demonstrate their correctness and efficiency. While prior familiarity with graphs isn't assumed, students should possess a level of "mathematical maturity," meaning they should be comfortable with proofs and abstract reasoning. The curriculum will encompass various topics, including graph basics, connectivity, matching, covering, planarity, sparsity, coloring, and network flows.

Objectives:

- Attain a comprehensive grasp of the foundational definitions and concepts within graph theory.
- Comprehend and apply the essential theorems and algorithms, generating illustrative instances as required and formulating subsequent inquiries.
- Develop adeptness in constructing proofs, employing elementary graph theory proof methodologies such as bijections, minimal counterexamples, and loaded induction.
- Cultivate the ability to articulate mathematical arguments within discussions and in written form clearly.
- Gain familiarity with graph theory's principal perspectives and objectives, including categorization, extremal analysis, optimization, precision, algorithms, and duality.
- Demonstrate the capability to employ graph theory knowledge to address challenges in diverse domains, potentially showcased through a class project.

Course Content:

Fundamental concepts of graphs: Basic definitions, graphs and digraphs, (induced) subgraphs, complementation, incidence and adjacency matrices, complete graphs, regular graphs, Petersen graphs, Handshaking lemma, bipartite graphs, Mantel's Theorem, Ramsey numbers, Turan's Theorem, and isomorphism.
Connectivity: Vertex and edge connectivity, cliques and independent sets, connected components, paths and cycles, cuts, blocks, k-connected graphs, Menger's theorem, diameter and shortest paths.
Trees and forests: Centers and centroids; spanning trees, Steiner trees, tree enumeration, Cayley's theorem. Traversability: Eulerian and Hamiltonian graphs, Dirac's theorem; Fleury's

algorithm for finding Eulerian paths or cycles, Traveling Salesman problem, Chinese Postman Problem.
Directed graphs: Tournaments, directed paths and cycles, Eulerian digraphs, connectivity and strongly connected digraphs, directed acyclic graphs (DAG), topological sorting.
Planarity: Plane and planar graphs, outerplanar graphs, maximal planar graphs, non-planarity of K5 and K3,3, Kuratowski's theorem, planar duals, Euler's formula, planar embeddings, genus, thickness, crossing number, planar separability theorem.
Matching and covering: Maximum matching, Berge's theorem, perfect matching, matching in bipartite graphs, Konig's theorem, Hall's marriage theorem, matching in general graphs, Tutte's theorem, weighted matching. Coloring: Vertex and edge coloring, clique number and chromatic number, vertex colouring, Brooks's theorem, Edge colouring, Vizing's theorem, 5-colour theorem for planar graphs, 4-color theorem for planar graphs, Wells-Powell algorithm for graph coloring.
Network flows: Flows and matching, max-flow min-cut theorem.
Graph classes: Intersection graphs, perfect graphs, interval graphs, comparability graphs, permutation graphs, chordal graphs, circle graphs, circular arc graphs, sparse graphs.
Selected topics: Graphs minors and graph decompositions (pathwidth and treewidth).

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Understand the basics of graphs, directed graphs, weighted graphs and be able to relate them to practical computational problems.
CO 2	Use effectively algorithmic techniques to study basic parameters and properties of graphs.

CO 3	Rigorously and formally analyze graph algorithms, prove their correctness, and discover their complexity using mathematical properties of graphs.
CO 4	Apply advanced knowledge and learn state-of-the-art techniques for the design and analysis of graph algorithms.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1										
CO2				2								
CO3		3		1								
CO4			3									2

Textbook

Douglas B. West: Introduction to Graph Theory, Second Edition, Pearson, Singapore, 2000.
Arthur Benjamin, Gary Chartrand, and Ping Zhang: The Fascinating World of Graph Theory, Princeton University Press, 2015.
Frank Harary: Graph Theory, CRC Press, 2018 (originally published in 1969).
J. A. Bondy and U.S.R. Murty: Graph Theory, Springer, 2008.
5. Reinhard Diestel: Graph Theory, Springer, 2000.

Course Title: **Distributed Computing System**		Credits: 4.0
Course Code: 0613 5167/5267	Contact Hours: 4 hours/week	

Rationale:

Distributed systems have become central to how computers are used, from web applications to e-commerce to content distribution. Distributed computing help programmers aggregate the resources of many networked computers to construct highly available and scalable services. This course teaches the abstractions, design and implementation techniques that enable building fast, scalable, fault-tolerant distributed computing systems.

Objectives:

To introduce students to the fundamental problems, concepts, and approaches in the design and analysis of distributed computing systems.
To familiarize students with the stages of the distributed system design cycle, including system architecture, data and processes arrangements, naming, communication and coordination issues, existing distributed computing paradigms, techniques, and tools, and evaluating the effectiveness of distributed application systems for specific data, task, and user types.

Course Content:

Introduction: From networked systems to distributed systems; Design goals; classification of distributed systems. Architectures: Architectural styles; Middleware and distributed systems; Layered-system architectures; Symmetrically distributed system architectures; Hybrid system architectures. Processes: Threads; Virtualization; Networked user interfaces; Servers; Code migration; Communication: Layered Protocols; Remote procedure call; Message-oriented communication; Multicast communication. Coordination: Clock synchronization; Physical clocks; Logical clocks; Mutual exclusion algorithm; Election algorithms; Gossip-based coordination; Distributed event matching; Location systems. Naming: Names, identifiers and addresses; Flat naming; Structured naming; The Domain Name System; The Network File System; Attribute-based naming; Named-data networking. Consistency and replication: Reasons for replication; Data-centric consistency models; Client-centric consistency models; Replica management; Consistency protocols. Fault tolerance: Failure models; Process resilience: Failure masking and Consensus; Reliable client-server communications; Reliable group communications; Distributed commit; Recovery: Checkpointing and Message logging; Security: Security threats; policies and mechanisms; Design issues; Symmetric and asymmetric cryptosystems; Hash functions; Key management; Authentication; Trust in distributed systems; Authorization and access control; Monitoring. Cloud Computing: cloud infrastructures; virtualization; software defined networks and storage; cloud storage; programming models. Blockchain: Blockchain technology overview; Technological and Cryptographic Elements in Blockchain; Blockchain Platforms; Blockchain Applications; Limitations, Opportunities and Challenges of Blockchain.

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Explainthe fundamental problems, concepts, and approaches in the design and analysis of distributed computing systems.
CO 2	Understand and analyze distributed algorithms for synchronization, consistency, fault-tolerance, security, etc. and reason about the correctness of those algorithms
CO 3	Derive algorithms for special cases
CO 4	Design and implement moderately complex, fault tolerant and secured distributed systems of their own

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO 1	PO 2	PO3	PO 4	PO5	PO6	PO7	PO 8	PO9	PO1 0	PO1 1	PO1 2
CO1	3	3										
CO2	3	3	3									
CO3			2	2								
CO4				1	3							

Textbook

Andrew S. Tanenbaum & Maarten van Steen: Distributed Systems: Principles and Paradigms, 4th ed, 2023.
https://webcms3.cse.unsw.edu.au/COMP9243/20T3/
https://www.distributed-systems.net/

Course Title: **Data Mining and Information Retrieval**	Credits: 4.0
Course Code: 0613 5191/5291	Contact Hours: 4 hours/week

Rationale:

This course explores advanced techniques in data mining and information retrieval, focusing on extracting valuable insights from large datasets and efficiently retrieving relevant information from various sources. Students will learn about foundational concepts, algorithms, and practical applications through lectures, hands-on exercises, and projects.

Objectives:

- To develop a solid foundation in the core concepts of data mining and information retrieval.
- To gain hands-on experience by applying various data mining and information retrieval techniques to real-world datasets.
- To learn to evaluate the performance of different data mining and retrieval models using appropriate metrics, compare the strengths and limitations of various algorithms to make informed decisions in selecting the right approach for specific tasks.
- To explore ethical considerations and privacy concerns related to the field, understand the potential biases in algorithms and strategies for mitigating them, ensuring responsible and fair data analysis.
- To be able to apply learned techniques to a real-world problem, develop the ability to design, implement, and present a comprehensive data mining or information retrieval solution.

Course Content:

Introduction: Overview of data mining and information retrieval, Applications and real-world examples

Data Preprocessing and Exploration: Data cleaning, transformation, and integration, Handling missing data and outliers, Exploratory data analysis techniques

Information Retrieval Fundamentals: Retrieval models (Boolean, Vector Space, Probabilistic, and Language models), Indexing, Term weighting and relevance ranking, Evaluation metrics (Precision, Recall, F1-score, MAP, NDCG)

Data Mining Techniques: Association rule mining, clustering algorithms (k-means, hierarchical, DBSCAN), Classification techniques (Decision trees, Support Vector Machines, Neural Networks)

Text Mining and Natural Language Processing: Text preprocessing (tokenization, stemming, lemmatization), Named Entity Recognition and Part-of-Speech tagging, Sentiment analysis and text classification, Topic modelling, Text classification and document retrieval using NLP techniques

Web Mining and Social Media Analytics: Crawling and indexing web content, Link analysis and PageRank algorithm, Opinion mining in social media, Stream mining and online algorithms, Feature selection and dimensionality reduction (PCA, t-SNE, LLE), Ensemble methods (Bagging, Boosting, Random Forests)

Recommender Systems: Collaborative filtering and content-based recommendation, Matrix factorization techniques, Evaluation of recommender systems

Large-Scale Data Mining and Distributed IR: MapReduce framework and Hadoop ecosystem, Scalable algorithms, Parallel and distributed indexing techniques

Ethical and Privacy Considerations: Privacy-preserving data mining, Bias and fairness in data mining and IR, Ethical issues in collecting and using data

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Demonstrate a comprehensive understanding of the fundamental concepts, models, and techniques in data mining and information retrieval, distinguishing between different approaches and their applications.
CO 2	Apply analytical skills to preprocess and explore datasets, identifying data quality issues, outliers, and patterns.
CO 3	Design and implement algorithms for tasks using appropriate methodologies to solve real world problems.

CO 4	Evaluate and interpret the performance of data mining and information retrieval techniques using relevant evaluation metrics and interpret results, identify strengths and limitations.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2	3											
CO3	3											
CO4			3									

Textbook
Introduction to Information Retrieval by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, Cambridge University Press

Course Title: **Computer Vision**		Credits: 4.0
Course Code: 0613 5195/5295	Contact Hours: 4 hours/week	

Rationale:
This course aims to give the students an in-depth knowledge of the foundation and application of computer vision. It will assist the students in gaining skills and experience in geometric and three-dimensional visual perception. After completing the course, students can train modern computer vision models for visual applications.

Objectives:
To introduce the students to advanced computer vision concepts, such as the fundamentals of image formation, geometric techniques of image processing, and feature matching for potential research.
To provide the students with knowledge of image pre-processing and analyzing with efficient motion estimation and tracking tools.
To train the students to apply machine learning models to solve vision-related problems, such as image segmentation, image enhancement, image classification, and object detection.

Course Content:
Digital Image Processing: Image Formation, Image Filtering, Principal Component Analysis, Edge Detection, Corner Detection,Shape detection, Scale-Invariant Feature Transform (SIFT) to detect and describe local features in images. Geometric Techniques in Computer Vision: Image Transformations, Camera Projections, Camera Calibration, Depth from Stereo, Two View Structure from Motion, Object Tracking. Machine Learning for Computer Vision: Introduction to Machine Learning, Image Classification, Object Detection, Semantic Segmentation, Large Scale Image Search.

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Describe the scope of challenges and applications of computer vision
CO 2	Make use of geometric camera models and multiple view geometry.

CO 3	Undertake video analysis problems such as tracking and structure from motion.
CO 4	Analyze cognitive tasks including image classification, recognition, and detection.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			3									
CO2				1								
CO3		2										
CO4				1								

Textbook

Computer Vision: A Modern Approach (Second Edition) by David Forsyth and Jean Ponce.

DIPA: Computer Vision and Image Analysis, Fourth Edition, Scott E Umbaugh, CRC Press.

Course Title: **Advanced Bioinformatics**		Credits: 4.0
Course Code: 0613 5197/5297	Contact Hours: 4 hours/week	

Rationale:

In this course, students will learn a comprehensive understanding of the principles and applications of bioinformatics. Throughout the course, students will learn to use various bioinformatics tools and software to analyze and interpret genomic data. Students will use the command line to undertake the analysis of sequence data and then derive biological inference and interpretation from those data, using diverse programs and scripting. Topics include sequence quality control, mapping, variant calling, differential expression, and visualization; as applied to the whole genome, the epigenome, and the transcriptome.

Objectives:

- Introduce students to molecular and cellular biology fundamentals, genetics, gene regulation, mutation, sequencing, and evolution.
- To help students develop skills that will enable them to formulate precise computational problems from biological problems.
- To help students develop skills that will enable them to design algorithms for biological problems.
- To show students how to apply many of the primary predictive methods common in modern bioinformatics.
- To give the students practical and hands-on experience with modern bioinformatics tools and databases.
- To train the students in the basic theory and application of programs used for database searching, protein and DNA sequence analysis, prediction of protein function, and building phylogenetic trees, computer-aided drug discovery.

Course Content:

Introduction: Introduction and History of Bioinformatics, Central Dogma of Molecular Biology, Modern Genomics, Biological Databases. Sequence Database: GenBank, NCBI Gene & Genome Retrieval, Protein Retrieval Uniprot. Structure and Function Database: Introduction To Protein Structure, Protein Data Bank File Format, Retrieval Of Protein Structure From Pdb, Protein Interaction Database: String. Sequence Analysis: Sequence Alignment, Blast, Finding Similar Gene

Sequences, Finding Similar Protein Sequences, Clustal Omega, Gene Prediction Prokayotes: Easygene, Gene Prediction Vertibrates: Hmmgene, Gene Mark: Gene Prediction From Genome, Expasy Translate, Ncbi Conserved Domain Search, Interpro, Island Viewer, Amr, Comprehensive Antibiotic Resistance Database (Card), Virulentpred. Structure Analysis: Protein Structure Prediction, Homology Modeling, Threading, Ab-Initio Modeling, Swiss Model, Alpha Fold2, Raptorx, Secondary Structure Prediction, Stride, Jpred-Protein Secondary Structure Prediction Server, Iupred, Pdbsum, Chimera: Structure Visualization, Aligning Protein Structures, Generation Of Mutant Structures. Genomics: Genome Assembly, Genome Annotation, Next Generation Sequencing (NGS): Mapping of Reads From Resequencing and Calling of Genetic Variants, Gene Expression and Transcriptome Sequencing, Quality control, Differential gene and transcript expression analysis of RNA-seq reads, Epigenomics. Phylogeny: Phylogeny, Muscle – Multiple Sequence Comparison By Log-Expectation, Molecular Evolutionary Genetics Analysis (Mega), Mega Software, Jalview. Computer-Aided Drug Discovery: Introduction To Cadd, Molecular Docking Introduction, Binding Pocket Prediction Using Cast-P, Ligand Molecule Retrieval From Pubchem, Drugbank Online – Database For Drug And Drug Target Info, Swiss Design, Lipinski's Rule, Admetlab, Swissdock

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Integrate biological concepts with information technologies to study the biological system.
CO 2	Apply the theoretical and computational methodologies relevant to the mining of information from large datasets of biological origin.
CO 3	Combine theory and practice to solve research problems in bioinformatics, genomics and proteomics.
CO 4	Manipulate data on computer and understand of the algorithms that underlie the analysis tools that are used for essential research tasks.

Mapping of Course Learning Outcomes to Program Learning Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			3									
CO3		3										
CO4				3								

Textbook
Bioinformatics: Sequence and Genome Analysis by David W. Mount
Bioinformatics with Python Cookbook by Tiago Antao
Understanding Bioinformatics by Market Zvelebil, Jeremy O. Baum
Bioinformatics for Biologists by Pavel Pevzner and Ron Shamir.
Genomics and Bioinformatics: An Introduction to Programming Tools for Life Scientists by Tore Samuelsson

Course Title: **IoT and Embedded Systems**		Credits: 4.0
Course Code: 0714 5173/5273	Contact Hours: 4 hours/week	

Rationale:
This course focuses on the design of an IoT ecosystem which includes all the components that enable businesses, governments, and consumers to connect to their IoT devices. The IoT ecosystem consists of remotes, dashboards, networks, gateways, data storage, analytics, and security. Students will be using STM32 IoT discovery node and cloud to demonstrate IoT practical applications. Illustrations of 5G and an overview of the "Security of Connected Devices" laws enacted by several states in the U.S. will also be covered. At the end of the course participants will be able to design a prototype of an IoT solution using sensor data, communication protocol, cloud, safety, and security risks

Objectives:

- Define the term "Internet of Things"
- State the technological trends which have led to IoT
- Describe the impact of IoT on society
- Define what an embedded system is in terms of its interface
- Enumerate and describe the components of an embedded system
- Describe the interactions of embedded systems with the physical world
- Name the core hardware components most commonly used in IoT devices
- Describe the interaction between software and hardware in an IoT device
- Describe the role of an operating system to support software in an IoT device
- Explain the use of networking and basic networking hardwar
- Describe the structure of the Internet
- Describe the meaning of a "network protocol"
- Explain MANETs and their relation to IoT

Course Content:
Evolution and technologies used in IoT, IoT embedded system programming, sensors and components, IoT stacks and usage on sensors, Design principles and methodologies used in IoT systems, IoT cloud architecture, primarily Azure , Connectivity and programming of IoT device using wireless transport and MQTT protocol, Case study for cybersecurity, privacy risks, safety and reliability, Illustration and evolution of 5G myriad IoT applications including smart cities, water waste, and agriculture

Course Learning Outcomes: After the successful completion of the course, students will be able to:

CO 1	Identify different components of IoT and comprehend the complexity of the IoT stack
CO 2	Provisioning, control and management of embedded IoT devices in the cloud
CO 3	Analytics and management of sensor data
CO 4	Implement appropriate communication protocols and data format such as Json

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PO of the corresponding department

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3											
CO2			3									
CO3				3								
CO4					3							

Textbook
Adrian McEwen and Hakim Cassimally, ―Designing the Internet of Things‖, John Wiley and Sons Ltd, UK, 2014.
VijayMadisetti, ArshdeepBahga, ―Internet of Things (A Hands-on Approach), Universities Press, 2015.

Course Title: **Contemporary Course on Computer Science and Engineering I**	Credits: 4.0
Course Code: 0611/0612/0613/0619 5187	Contact Hours: 4 hours/week

This course covers a contemporary title in Computer Science and Engineering decided by the department.

Course Title: **Contemporary Course on Computer Science and Engineering II**	Credits: 4.0
Course Code: 0611/0612/0613/0619 5287	Contact Hours: 4 hours/week

This course covers a contemporary title in Computer Science and Engineering decided by the department.

D. **Doctoral by Research**

Credit requirements and duration of Doctoral Program are as follows.

Credit Requirement			Program Duration			
Coursework (Min.)	Dissertation (Min.)	Total (Min.)	Semester (Min.)	Year (Min.)	Semester (Max.)	Year (Max.)
Non-credit (if offered)	72	72	6.0	3.0	12	6.0

The course structure for this program is given below.

Semester I

Course Code	Course Title	Hours/Week		Credits
		Theory	Lab	
CSE 0610 7670	Independent Study	0	36	12.00
Total		36		12.00

Semester II

Course Code	Course Title	Hours/Week		Credits
		Theory	Lab	
CSE 0610 7670	Independent Study (continued)	0	36	12.00
Total		36		12.00
Grand Total				24.00

Semester III

Course Code	Course Title	Hours/Week		Credits
		Theory	Lab	
CSE 0610 7670	Independent Study (continued)	0	36	12.00
Total		36		12.00
Grand Total				36.00

Semester IV

Course Code	Course Title	Hours/Week		Credits
		Theory	Lab	
CSE 0610 7670	Independent Study (continued)	0	36	12.00
	Total	**36**		**12.00**
	Grand Total			**48.00**

Semester V

Course Code	Course Title	Hours/Week		Credits
		Theory	Lab	
CSE 0610 7670	Independent Study (continued)	0	36	12.00
	Total	**36**		**12.00**
	Grand Total			**60.00**

Semester VI

Course Code	Course Title	Hours/Week		Credits
		Theory	Lab	
CSE 0610 7670	Independent Study (continued)	0	36	12.00
	Total	**36**		**12.00**
	Grand Total			**72.00**