The National Institute of Engineering, Mysuru

Dept. of CSE - NEP Batch (2021 admitted) Scheme of Teaching – Semester 3 to 8

III Semester

Sl. No	Course Code	Course Title	Category	Teaching Department	Teaching Hours / Week				Examination		Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Self-study Component			
					L	T	P	S	CIE Marks	SEE Marks	
1	21MA3C03	Transforms, Optimization and Number Theory	BSC	Maths	3	0	0	0	50	50	3
2	21CS3C01	Data Structures	PC	CSE	3	0	0	0	50	50	3
3	21CS3C02	Discrete Mathematical Structures	PC	CSE	3	2	0	0	50	50	4
4	21CS3C03	Computer Organization and Architecture	PC	CSE	3	2	0	0	50	50	4
5	21CS3C04	Logic Design	PC	CSE	3	0	0	0	50	50	3
6	21CS3L01	Data Structures Laboratory	PC	CSE	0	0	2	0	25	25	1
7	21CS3L02	Logic Design Laboratory	PC	CSE	0	0	2	0	25	25	1
8	21BG3C03	Biology for Engineers	PC	BSC	2	0	0	0	50	50	2
9	21KA3H01/ 21CI3H01	Kannada/ CIPE	HSC	Hum.	-	-	-	-	50		0
10	21CS3AXX	AEC3	AEC	CSE	1	0	0	0	50	0	1
11	21CS3U01	Social Connect and Responsibility	UHV	CSE	-	-	-	--	-	-	0
TOTAL											**22**

Lateral Entry Students

12	21MA3N01	Additional Mathematics 1	BSC	Maths	2	2	0	0	50	-	0

The National Institute of Engineering, Mysuru

Dept. of CSE - NEP Batch (2021 admitted) Scheme of Teaching – Semester 3 to 8

IV Semester											
Sl. No	Course Code	Course Title	Category	Teaching Department	Teaching Hours / Week				Examination		Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Self-study Component			
					L	T	P	S	CIE Marks	SEE Marks	
1	21MA4C03	Statistics and Probability	BSC	Maths	3	0	0	0	50	50	3
2	21CS4C01	Design and Analysis of Algorithms	PC	CSE	3	0	2	0	50	50	4
3	21CS4C02	Formal Languages and Automata Theory	PC	CSE	3	2	0	0	50	50	4
4	21CS4C03	Object Oriented Programming	PC	CSE	3	0	0	0	50	50	3
5	21CS4C04	Software Engineering	PC	CSE	3	0	0	0	50	50	3
6	21CS4L01	Object Oriented Programming Laboratory	PC	CSE	0	0	2	0	25	25	1
7	21KA4H01/ 21CI4H01	Kannada / CIPE	HSC	Hum.	-	-	-	-	50		0
8	21CS4AXX	AEC4	AEC	CSE	1	0	0	0	50	0	1
9	21CS4U01	Universal Human Values and Professional Ethics	UHV	CSE	1	0	0	0	50	0	1
10	21CS4I01	Summer Internship 1	PI	Completed during II Sem Vacation					50	0	2
TOTAL											22

Lateral Entry Students

11	21MA4N01	Additional Mathematics 2	BSC	Maths	2	2	0	0	50	-	0

The National Institute of Engineering, Mysuru

Dept. of CSE - NEP Batch (2021 admitted) Scheme of Teaching – Semester 3 to 8

V Semester											
Sl. No	Course Code	Course Title	Category	Teaching Department	Teaching Hours / Week				Examination		Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Self-study Component			
					L	T	P	S	CIE Marks	SEE Marks	
1	21CS5C01	Operating Systems	PC	CSE	3	0	0	0	50	100	3
2	21CS5C02	Microprocessor and Interfacing	PC	CSE	3	0	2	0	50	100	4
3	21CS5C03	Database Management System	PC	CSE	3	0	0	0	50	100	3
4	21CS5C04	Software Project Management and Entrepreneurship	PC	CSE	3	2	0	0	50	100	4
5	21CS5C05	Introduction to Data Science	PC	CSE	2	0	0	0	50	50	2
6	21CS5L01	Database Laboratory	PC	CSE	0	0	2	0	25	25	1
7	21CS5OXX	Open Elective 1	OE	Other Dept.	3	0	0	0	50	100	3
8	21CS5A01	Research Methodology and IPR	AEC	CSE	2	0	0	0	50	50	2
9	21HS5C01	Physical Education/Yoga & NSS	HSC	Physical Education	-	-	-	-	50	0	0
TOTAL											22

VI Semester

Sl. No	Course Code	Course Title	Category	Teaching Department	Teaching Hours / Week				Examination		Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Self-study Component			
					L	T	P	S	CIE Marks	SEE Marks	
1	21CS6C01	Artificial Intelligence and Machine Learning	PC	CSE	3	0	0	0	50	100	3
2	21CS6C02	Computer Networks	PC	CSE	3	0	0	0	50	100	3
3	21CS6C03	Fundamentals of DevOps	PC	CSE	3	0	0	0	50	100	3
4	21CS6C04	Full Stack Development	PC	CSE	3	0	2	0	50	100	4
5	21CS6L01	Computer Networking Laboratory	PC	CSE	0	0	2	0	25	25	1
6	21CS60XX	Open Elective 2	OE	Other Dept.	3	0	0	0	50	100	3
7	21CS6C05	Minor Project	PC	CSE	0	0	4	0	50	50	2
8	21CS6AXX	AEC 6	AEC	CSE	1	0	0	0	50	0	1
9	21HS6C01	Environment Studies	HSC	Civil					50		0
10	21HS6C02	Physical Education/Yoga & NSS	HSC	Physical Education					50	0	0
11	21CS6I01	Summer Internship 2	PI	Completed during IV Sem Vacation					50	50	2
TOTAL											**22**

The National Institute of Engineering, Mysuru

Dept. of CSE - NEP Batch (2021 admitted) Scheme of Teaching – Semester 3 to 8

VII Semester

Sl. No	Course Code	Course Title	Category	Teaching Department	Teaching Hours / Week				Examination		Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Self-study Component			
					L	T	P	S	CIE Marks	SEE Marks	
1	21CS7E1X	Elective-1	PC	CSE	3	0	0	0	50	100	3
2	21CS7E2X	Elective-2	PC	CSE	3	0	0	0	50	100	3
3	21CS7E3X	Elective-3	PC	CSE	3	0	0	0	50	100	3
4	21CS7E4X	Elective-4	PC	CSE	3	0	0	0	50	100	3
5	21CS7AXX	AEC 6 - MOOC	AEC	CSE	0	0	2	0	-	100	3
6	21SC7S01	Technical Seminar	PI	CSE	0	0	0	2	50	-	1
TOTAL									**250**	**500**	**16**

The National Institute of Engineering, Mysuru

Dept. of CSE - NEP Batch (2021 admitted) Scheme of Teaching – Semester 3 to 8

VIII Semester

Sl. No	Course Code	Course Title	Category	Teaching Department	Teaching Hours / Week				Examination		Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Self-study Component			
					L	T	P	S	CIE Marks	SEE Marks	
1	21CS8P01	Project Work (to be completed in VII Semester)	PI	CSE	0	0	8	4	100	100	8
2	21CS8I01	Research/Industry Internship	PI	CSE	-	-	-	-	100	100	8
TOTAL									**200**	**200**	**16**

III SEMESTER

Course: Transforms, Optimization & Number Theory

(CSE/ISE)

Course Code: 21MA3C03

Credits: 3 **L:T:P:3:0:0**

SEE: 50% Marks **CIE: 50% Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	Calculus and basics of Number theory
Learning objectives	The goal of this course is **I**. To enable the students to find the sum of infinite series using Fourier series and finding certain integrals using Fourier transforms **II**. To enable the students to find the optimal solution of Linear Programming Problems **III**. To facilitate the students to apply the concepts of number theory in cryptography

Course Outcomes:

On the successful completion of the course, the student will be able to

COs		**Bloom's level**	**Expected attainment level %**
CO1	Translate the periodic function of period $2l$ in terms of Fourier series, half range series.	Apply	60
CO2	Compute Fourier and Inverse Fourier transforms of functions	Apply	60
CO3	Use optimization techniques to solve Linear Programming problems	Apply	60
CO4	Solve problems of linear congruences and its applications to cryptography	Apply	60

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	S	M										L			
CO2	S	M										L			
CO3	S	M										L			
CO4	S	M										L			

S – Strong (3) M – Medium(2) L – Low (1)

Course Structure

	Module – 1: Fourier Series	**No. of Lecture Hours**	**No. of Tutorial Hours**
1.1	Periodic functions, Fourier series, Dirichlet’s conditions for a Fourier series, Euler’s Fourier coefficients.	1	Nil
1.2	Fourier series of even and odd functions of period $2l$– continuous and discontinuous functions	4	Nil
1.3	Half range Sine and Cosine series	2	Nil
1.4	Practical harmonic analysis	1	Nil
	Module-2: Fourier transforms		
2.1	Infinite Fourier Transforms	2	Nil
2.2	Fourier cosine and sine transforms	2	Nil
2.3	Inverse cosine andsine Transforms	2	Nil
2.4	Discrete Fourier Transforms, properties	2	Nil
	Module-3: Optimization		
3.1	Standard form of LPP, Graphical method	2	Nil
3.2	Simplex method	2	Nil
3.3	application problems	2	Nil
3.4	Duality, degeneracy	2	Nil
	Module-4: Number Thoery 1		
4.1	Euclidean Algorithm	2	Nil
4.2	solution of linear Diaphantine equations	2	Nil
4.3	solution of linear congruence	1	Nil
4.4	Chinese Remainder theorem-problems	1	Nil
4.5	Fermat's little theorem-problems	1	Nil
4.6	Wilson's theorem-problems	1	Nil
	Module-5: Number Theory 2		
5.1	Euler’s theorem -problems	1	Nil
5.2	Basic Group Theory - Groups, subgroups	2	Nil
5.3	cyclic groups	2	Nil
5.4	Applications to cryptography: Caesar cipher, Multiplicative cipher.	3	Nil
	Total No. of Lecture Hours	40	
		Total No. of Tutorial Hours	Nil

Text Books:

1. Higher Engineering Mathematics – B.S. Grewal, 42nd edition, Khanna Publications.
2. Elementary Number Theory -David Burton, 7th edition, Mc Graw Hill publication

Reference Books :

1. Advanced Engineering Mathematics – H. K. Dass, Chand Publications.
2. Higher Engineering Mathematics – B. V. Ramana, Tata McGraw-Hill Publications.
3. Advanced Engineering Mathematics - Erwin Kreyszig, wiley publications, 10th edition.
4. Operations Research-An introduction by Hamdy A Taha-10th edition, Pearson.

Course: Data Structures

Course Code:21CS3C01

Credits: 3 **L:T:P: 3:0:0**

CIE: 50% Marks **SEE: 50% Marks**

SEE Hours: 3 Hours **Max. Marks: 100**

Prerequisites if any	**Problem Solving through Programming**
Learning objectives	1. To provide the knowledge to explain the fundamentals of data structures and their applications essential for problem solving. 2. To acquire the knowledge on applying the concepts of Linked lists for solving the real-world problems. 3. To understand linear representation of data structures: Stack, and Queue. 4. To acquire the knowledge to solve real-world problems using non-linear data structures like trees. 5. To gain the knowledge for applying the sorting algorithms and hash functions to solve real-world problems.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Apply the concepts of pointers and structures in problem solving.	L3
CO2	Use different types of linked lists to solve problems.	L3
CO3	Demonstrate stack and queue data structures to solve problems.	L3
CO4	Illustrate the operations performed on tree data structures.	L3
CO5	Apply different sorting methods and hash functions for problem solving.	L3

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1	3	2	2									2		1	1
CO2	3	2	3									2		3	3
CO3	3	2	3									2		3	3
CO4	3	2	3									2		3	3
CO5	3	2	3									2		2	2

Mapping Strength : 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	**Modules**	**No. of Lecture Hours**	**No. of Tutorial Hours**
	Module – 1	**6**	
1.1	**Pointers:** Introduction	1	0
1.2	Structures, Nested Structures, Arrays of Structures	2	0
1.3	Structures and Functions, Self-Referential Structures	2	0
1.4	**Introduction to Data Structures,** Classification of data structures, Operations on data structures, Abstract data type	1	0
	Module – 2	**9**	
2.1	**Linked Lists -** Singly linked lists	3	0
2.2	Circular linked lists	2	0
2.3	Doubly linked lists	2	0
2.4	Circular doubly linked lists	2	0
	Module – 3	**9**	
3.1	**Stacks**: Introduction, Array Representation of Stacks	1	0
3.2	Operations on a Stack, Linked Representation of Stacks	1	0
3.3	**Applications of Stacks:** Conversion of an infix expression into a postfix expression, Evaluation of a postfix expression, Recursion.	3	0
3.4	**Queues:** Introduction, Array Representation of Queues, Linked Representation of Queues	2	0
3.5	**Types Of Queues:** Circular Queue, Priority Queues	2	0
	Module – 4	**9**	
4.1	**Trees:** Introduction, Types of Trees	1	0
4.2	**Traversing** A Binary Tree,	1	0
4.3	Binary Search Trees, Operations on Binary Search Trees	3	0
4.4	**Threaded Binary Trees:** One-way Threading,	1	0
4.5	AVL Tree,Binary Heaps operations	3	0
	Module – 5	**7**	
5.1	**Sorting:** Introduction to Sorting , Radix Sort , Heap Sort	2	0
5.2	**Hashing and Collision:** Introduction, Hash Tables, Different Hash Functions	2	0
5.3	**Collisions**	2	0
5.4	Pros and Cons of Hashing, Applications of Hashing	1	0
	Total No. of Lecture Hours	40	
		Total No. of Tutorial Hours	0

Text Books:

1. ReemaThareja, “Data Structures using C”, 2nd Edition, 2018, Oxford University Press.

Reference Books:

1. Aaron M Tenenbaum, YedidyahLangsam and Moshe J Augenstein, “**Data Structuresusing C**”, 2014, low price edition,Pearson education,.
2. Richar F Gilberg and Behronz A Forouzan, “**Data Structures, A Pseudocode Approach with C”**, 2nd Edition,2012, Thomson.
3. Horowitz, Sahni, Anderson-Freed, “**Fundamentals of Data Structures in C**”,2nd Edition,2011, Universities Press.

Online Resources:

1. NPTEL: Programming and Data structures- https://nptel.ac.in/courses/106/105/106105085/
2. Coursera: Data Structures - https://www.coursera.org/learn/data-structures
3. Programming & Data structures: http://nptel.ac.in/courses/106106130/
4. Programming, Data structures and Algorithms: http://nptel.ac.in/courses/106106133/

Course: Discrete Mathematical Structures

Course Code: 21CS3C02

Credits: 4 **L:T:P:S 3:2:0:0**

CIE: 50% Marks **SEE: 50% Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	**Mathematics**
Learning objectives	1. Solve problems using concepts of Functions. 2. Solve problems using Relations and its properties. 3. Understand Generating Functions and Recurrence Relations and their applications in computers 4. To introduce concepts and properties of Graphs 5. To introduce the concepts of Trees and its properties.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Use Functions and its concept in solving real world problem.	L2, L3
CO2	Apply the concept of Relations and its properties in solving problems.	L2, L3
CO3	Solve problems using Recurrence Relations and Study its applications in computers.	L2, L3
CO4	Solve problems using concepts of graphs and analyze its real world applications.	L2, L3
CO5	Synthesis tree structure paradigm.	L2, L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	2					2	1	2	2		1	2
CO2	3	3	2	2					2	1	2	2		2	3
CO3	3	3	2	2					2	1	1	2		2	3
CO4	3	3	2	3					2	1	3	2		2	3
CO5	3	3	2	3					2	1	3	2		2	3

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	**Modules**	**No. of Lecture Hours**	**No. of Tutorial Hours**
	Module – 1		
1.1	**Functions:** Cartesian Products and Relations	1	0
1.2	Plain and One-to-One, Onto Functions	2	0
1.3	The Pigeonhole Principle	2	0
1.4	Function Composition and Inverse Functions	2	0
T	Solve problems using concepts of Functions and Pigeonhole Principle	0	2
	Module – 2		
2.1	**Relations:** Properties of Relations,	2	0
2.2	Computer Recognition – Zero-One Matrices and Directed Graphs	2	0
2.3	Partial Orders – Hasse Diagrams.	3	0
T	Solve problems using concepts of Relation and Partial orders	0	1
	Module – 3		
3.1	**Recurrence Relations:** First order linear recurrence relations,	2	0
3.2	The Second order linear homogeneous recurrence relation with constant coefficients	2	0

3.3	Non Homogeneous recurrence relation	3	0
T	Analysis of Application of Recurrence Relations in Computers (calculating time complexity)	0	2
	Module – 4		
4.1	**Graph Theory and Applications**: Definitions and Examples Sub graphs, Complements	2	0
4.2	Graph Isomorphism, Vertex Degree, Euler Trails and Circuits	2	0
4.3	Planar Graphs	2	0
T	Application of Graphs in Computers	0	1
	Module – 5		
5.1	Hamilton Paths and Cycles	2	0
5.2	Graph Coloring, and Chromatic Polynomials	2	0
5.3	**Trees:** Definitions, Properties, and Examples	3	0
5.4	Routed Trees	2	0
5.5	Trees and Sorting	2	0
5.6	Weighted Trees and Prefix Codes	2	0
T	Solve problems on Hamilton paths,Graph Coloring and Polynomials and activity to study their applications. Activity to study the application of Trees in Computers	0	4
Total No. of Lecture Hours		**40**	
Total No. of Tutorial Hours			**10**

Text Books:

1. **Discrete and Combinatorial Mathematics,** Ralph P. Grimaldi, 5th Edition, PHI/Pearson Education, 2004.

Reference Books:

1. **Handbook of discrete and combinatorial mathematics**, Kenneth H.Rosen, John G.Michels.
2. **Mathematics of Computer Science**, Prof. Albert R.Meyer, MIT Open Course Ware.
3. **Concrete Mathematics: A foundation for computer science**, Ronald L.Graham, Donald Ervin Knuth, Oren Patashnik
4. **Graph Theory with Applications to Engineering and Computer Science** by NarsinghDeo, Prentice-Hall, 2004

Course: Computer Organization and Architecture

Course Code: 21CS3C03

Credits: 4 **L:T:P - 3:2:0**

CIE: 50% Marks **SEE: 50% Marks**

SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	**Basic concepts in digital circuit design, Familiarity with a programming language like C or C++**
Learning objectives	This course is intended to teach 1. The basic structure of computers and instruction set architecture. which includes the general concepts of Stacks and subroutine. 2. The basics involved in number representation and arithmetic operations in the computer system. 3. Basic processor concept ,instruction execution and Bus architecture 4. Instruction level parallelism ,Dependences and hazards 5. Memory architecture and mapping techniques. 6. Input output techniques and Interrupts

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Describe an overview of computer hardware and software and Encode machine Instructions.	L2, L3
CO2	Use algorithms to perform fast multiplication, division and to represent floating point numbers in binary.	L2, L3
CO3	Explain the basic processing unit and design of control system	L2
CO4	Describe the importance of pipelining to achieve instruction level parallelism and solve problems on data dependences and hazards.	L2, L3
CO5	Describe memory architecture and mapping techniques,	L2
CO6	Explain the basics concepts of I/O, interrupts, direct memory access technique and types of busses.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2						2	2		1			1
CO2	3	2	2						2	2		1			3
CO3	3	2	2						2	2		1			3
CO4	3	2	2						1	2		1			3
CO5	3	2	2						1	2		1			2
CO6	3	2	2						1	2		1			2

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

SL No	Modules	No. of Lecture Hours	No. of Tutorial Hours	Self Learning Hours
	Module – 1			
1.1	Functional units**,** bus structures, performance,	1	0	0
1.2	Memory locations and addresses, Memory operations, instructions and instruction sequencing,	2	0	0
1.3	Encoding of machine instructions	2		
1.4	Stacks and Queues, Subroutines	2	0	0
1.5 (T)	Problems on Numbers, Arithmetic operations, characters and Encoding of machine instructions	0	2	0
	Module – 2			
2.1	**Arithmetic unit:** Multiplication of Positive numbers	2	0	0
2.2	A signed operand multiplication ,Bit pair recoding of multipliers	3	0	0
2.3	Integer division, IEEE standard for floating point numbers.	2	0	0
2.4 (T)	Booth algorithm and its Implementation Problems on fast multiplication	0	2	0
	Module – 3			
3.1	**The Processor:** Fundamental concepts	1	0	0
3.2	Execution of complete instruction, Multiple-Bus	2	0	0

	organization			
3.3	Hardwired control unit and Micro programmed control unit	4	0	0
3.4 (T)	Design of Hardwired control unit and Micro programmed control unit	0	2	0
	Module – 4			
4.1	**Instruction level parallelism:** Concepts and challenges, Data dependences and Hazards: Data dependences, Name dependences, Control Dependences	2	0	0
4.2	**Pipelining**: Introduction, A simple implementation of a RISC instruction set, The classic five-stage pipeline for a RISC processor,	2	0	0
4.3	**The major hurdle of pipelining-** pipeline hazards, Performance of pipeline stages, structural hazards, Data hazards, Branch hazards.	2	0	0
4.4 (T)	Problems on data dependences and hazards, Basic performance issues in pipelining	0	2	0
	Module – 5			
5.1	Memory unit:Basic concepts ,Internal organisation of memory chips ,Structure of larger memory	2	0	0
5.2	Cache memories ,Mapping functions	3	0	0
5.3	Replacement Algorithms	2	0	0
5.4	**Input Output Unit**: Accessing I/O devices, **Interrupts:** Interrupt hardware, Enabling and disabling of interrupts	3	0	0
5.5	Direct Memory Access –Bus arbitration, Synchronous bus and asynchronous bus.	3	0	0
5.6 (T)	Problems on mapping techniques, Demonstrating I/O instruction and interupts	0	2	0
	Total No. of Lecture Hours	**40**		
	Total No. of Tutorial Hours		**10**	
	Total No. of Self learning Hours			**0**

Text Books:

1. C Hamacher, Z Vranesic, S Zaky: **Computer Organization,** Tata McGraw Hill, 5th Edition, 2011.

2. John L Hennessy, David A Patterson: **Computer Architecture A Quantitative Approach**, Elsevier, 5th Edition 2012.

Reference Books:

1. David A. Patterson, John L. Hennessy: **Computer Organization and Design**, M.K Publishers, 4th edition, 2010

2. William Stallings, **"Computer Organization and Architecture, Designing for Performance"**, 9e, Pearson, 2014.

Online Resources:

1. **NPTEL:** Computer Organization and Architecture A Pedagogical Aspect
   https://onlinecourses.nptel.ac.in/noc19_cs04/preview

2. Edx: Computation Structures 3:Computer Organization
   https://www.edx.org/course/computation-structures-
   3-computer-mitx-6-004-3x-0

Course: Logic Design

Course Code: 21CS3C04

Credits: 3 **L:T:P : 3:0:0**

CIE: 50% Marks **SEE: 50% Marks**

SEE : 3 Hours **Max. Marks:100**

Prerequisites if any	**Fundamentals of Logic**
Learning objectives	1. To provide the knowledge to explain the fundamentals of Logic circuits and combinational circuits. 2. To introduce the design of sequential circuit systems. 3. To provide the knowledge to explain the working of Registers and design counters. 4. To introduce the working of Memory and Programmable logic.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Apply logic minimization techniques to design digital circuits using logic gates.	L3
CO2	Design combinational circuits using logic gates.	L3
CO3	Design and analyze sequential circuits systems.	L3
CO4	Discuss the design of Registers and Counters.	L3
CO5	Describe the design of Memory and Programmable logic.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	3	2	1	1				1	1		1			2
CO2	3	3	2	1	1				1	1		1			2
CO3	3	3	2	1	1				1	1		1			2
CO4	3	3	2	1	1				1	1		1			2
CO5	3	3	2	1	1				1	1		1			2

Mapping Strength : 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No. of Lecture Hours	No. of Tutorial Hours
	Module – 1	**8**	
1.1	**Boolean Algebra**: Introduction, Basic Definitions, Basic Theorems and Properties, Boolean Functions, Canonical Standard Forms.	2	0
1.2	**Simplification of Boolean Functions**: The Map Method, Two-and Three-Variable Maps, Four- Variable Maps	2	0
1.3	Product of Sums Simplification, Don't-Care Conditions	2	0
1.4	The Tabulation Method, Determination of Prime-Implicants, Selection of Prime-Implicants.	2	0
	Module – 2	**8**	
2.1	**Combinational Logic**: Introduction, Design Procedure	1	0
2.2	Adders, Subtractors	1	0
2.3	Binary Parallel Adder, Decimal Adder,	2	0
2.4	Magnitude Comparator, Decoders	2	0
2.6	Multiplexers.	2	0

	Module – 3	**9**	
3.1	**Sequential Logic**: Introduction, Flip-Flops, Triggering of Flip-Flops,	2	0
3.2	Analysis of Clocked Sequential Circuits,	3	0
3.3	State Reduction and Assignment,	2	0
3.4	Design Procedure	2	0
	Module – 4	**8**	
4.1	**Registers, Counters**: Introduction, Registers,	2	0
4.2	Shift Registers, Ripple-Counters,	2	0
4.3	Synchronous Counters.	2	0
4.4	Ring Counter,Johnson Counter.	2	0
	Module – 5	**7**	0
5.1	**Memory and Programmable Logic:** Introduction	1	0
5.2	Random-access memory	2	0
5.3	Read-only Memory	2	0
5.4	,Programmable Logic array, Programmable array Logic.	2	0
	Total No. of Lecture Hours	**40**	
		Total No. of Tutorial Hours	**-**

Text Books:

1. **Digital Design:** M. Morris Mano, Pearson (2013), new print Fifth edition

Reference Books:

1. **The Elements of Computing System – Building the Modern Computer from First Principles:** Noam Nisan, Shimon Schocken, The MIT Press (2005).

2. **Digital Principles and Applications:**Donald P Leach, Albert Paul Malvino & Goutham Saha, TMH, 6th Edition, 2006**.**

3. **Fundamentals of Digital Logic with Verilog Design:** Stephen Brown, Zvonko Vranesic, TMH, 2006

4. **Fundamentals of Logic Design: Charles** H. Roth, Jr., Thomson, 5th Edition, 2004

Online Resources:

1. NPTEL: Programming and Data structures- https://onlinecourses.nptel.ac.in/noc21_cs64
2. Coursera: Digital Systems - https://www.coursera.org/learn/digital-systems

Course: Data Structures Laboratory

Course Code: 21CS3L01

Credits: 1 L:T:P- 0:0:2

CIE: 50% Marks SEE: 50% Marks

SET Hours: 2 Max. Marks:25

Prerequisites if any	Fundamentals of C programming
Learning objectives	1. To gain the knowledge to apply the advanced C concepts like pointers and structures. 2. To acquire knowledge to implement the linear data structures such as stack, queue and linked list to solve the real world problems 3. To acquire knowledge to implement the non linear data structures like trees to solve the real world problems 4. To gain the knowledge to apply the

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Apply the advanced C concepts like pointers, structures, dynamicmemory allocation for problem solving.	L3
CO2	Demonstrate different types of linked lists and use them to develop applications.	L3
CO3	Illustrate stack data structure for solving problems.	L3
CO4	Solve the problems using queue data structure.	L3
CO5	Apply binary tree data structure, sorting and hashing methods to solve problems.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	2	2					1				2		1	1
CO2	3	2	3					1				2		3	3
CO3	3	2	3					1				2		3	3
CO4	3	2	3					1				2		3	3
CO5	3	2	3					1				2		2	2

Mapping Strength : 3 – Strong 2 – Medium 1 – Low

List of Experiments

Sl.No	Experiment	Hands on/ Virtual
1	(a)Write a program to accept 3 integers and find the maximum among 3 numbers using functions and pointers. (b) Write a C program using pointer for searching the desired element from the array using pointers. (c) Write a program to find the maximum element in each row of the matrix using pointers.	Hands on
2	(a) Write a C program to read and display the Time in specified format. Create a structure called TIME with hour(int), minute(int), second(int) and next(self-referencing pointer) as its members. Dynamically create two variables of structure TIME and link the first variable to the second one and display it. Write a Display function that takes address of first TIME variable and displays both times in the format h:m:s. (b) Write a C program to read and display the student details. Define a structure 'Student' with fields name(string) , usn(int), marks of 3 subjects (int) and average(float) in it. Store the details of n students in an array of structure 'Student'. Display the details of all students in the descending order their total marks.	Hands on
3	(a) Write a C program using dynamic variables and pointers, to construct a singly linked list. The operations to be supported are: i) Insert at the front of a list. ii) Insert at any position in the list. iii) Deleting a node based on specified value. iv) Searching a node based on specified value. v) Displaying all the nodes in the list. b) Write a C program to reverse a linked list elements.	Hands on
4	Write a C program to support the following operations on a doubly linked list where each node consists of integers. i) Create a doubly linked list by adding each node at the front. ii) Insert a new node to the left of the node whose key value is read as an input. iii) Delete the node of a given data, if it is found, otherwise display appropriate message. iv) Display the contents of the list.	Hands on

5	Write a program to design, Develop and Implement a menu-driven program in C for the following operations on STACK of integers (Array implementation of the stack with maximum size MAX = 4). i) Push an element on to stack. ii) Pop an element from the stack. iii) Check Overflow and Underflow situations on the stack. iv) Display the contents of stack. v) Exit. Support the program with appropriate functions for each of the above operations.	Hands on
6	a) Write a C program to convert an expression given in "infix" form to "postfix" form using stack concept. b) Check whether a given string is a palindrome or not by using a stack. c) Write a program to find the nth term in the Fibonacci series using recursion.	Hands on
7	a) Write a C program to simulate the working of a queues using an array provide and implement the following operations: i) Insert ii) Delete iii) Display Assume that the size of the queue is 5. b) Write a C program to implement a circular queue using linked lists.	Hands on
8	a) Write a C program to construct a binary search tree of integers and also display the elements in the tree using Inorder, Preorder and Postorder traversals. b) Write a C program to find the number of leaf nodes in a BST c) Write a C program to print all root to leaf paths of a BST.	Hands on
9	a) Write a C program to sort set of integers using radix sorting technique. b) Write a C program to search using closed hashing.	Hands on

Text Books:

1. ReemaThareja, **"Data Structures using C",** 2nd Edition, 2018, Oxford University Press.

Reference Books:

1. Aaron M Tenenbaum, YedidyahLangsam and Moshe J Augenstein, **"Data Structures using C",** 2014, low price edition,Pearson education,.
2. Richar F Gilberg and Behronz A Forouzan, **"Data Structures, A Pseudocode Approach with C",** 2nd Edition,2012, Thomson.

3. Horowitz, Sahni, Anderson-Freed, **"Fundamentals of Data Structures in C",** 2nd Edition,2011, Universities Press.

Online Resources:

1. NPTEL: Programming and Data structures- https://nptel.ac.in/courses/106/105/106105085/
2. Coursera: **Data Structures -** https://www.coursera.org/learn/data-structures
3. **Programming & Data structures:** http://nptel.ac.in/courses/106106130/
4. **Programming, Data structures and Algorithms:** http://nptel.ac.in/courses/106106133/

Course: Logic Design Laboratory

Course Code: 21CS3L02

Credits: 1 **L:T:P - 0:0:2**

CIE: 50% Marks **SEE: 50% Marks**

SET Hours: 2 **Max. Marks:25**

Prerequisites if any	**Fundamentals of Logic Design**
Learning objectives	1. To provide the knowledge to explain the fundamentals of Logic circuits and combinational circuits. 2. To introduce the design of sequential circuit systems. 3. To provide the knowledge to explain the working of Registers and design counters. 4. To provide the knowledge to simulate combinational and sequential circuits using simulators.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Illustrate the working of logic gates.	L3
CO2	Design Combinational circuits.	L3
CO3	Synthesize Sequential circuits.	L3
CO4	Synthesize combinational and sequential circuits using simulators.	L3

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1	3	2	2	1	1				2	2		1			3
CO2	3	2	2	1	1				2	2		1			3
CO3	3	2	2	1	1				2	2		1			3
CO4	3	2	2	1	1				2	2		1			3

Mapping Strength : 3 – Strong 2 – Medium 1 – Low

List of Experiments

PART-A

HARDWARE

1. Understand the working of Basic gates and implementation of Basic gates using universal gates .

2. a) Implement Half adder and Full adder using logic gates.
   b) Implement Half subtractor, Full subtractor using logic gates.

3. Implement BCD to Excess-3 using basic gates.

4. a) Realize 3:8 decoder using logic gates.
b) Realize a full adder or any Boolean function using 3:8 decoder IC.

5. Give any four variable logic expression, realize the simplified logic expression using 8:1 multiplexer.

6. Realize a JK master Slave Flip- Flop using NAND gates and verify its truth table.

7. Realize mod-4 and mod-6 counter using Synchronous counter design.

8. Design an asynchronous counter mod 10,mod 6 and mod 7 from a Decade counter .

9. Realize Ring counter and Johnson counter.

PART-B

SIMULATION USING XILINX

1. Write a VHDL code for a Half –Adder ,Simulate and verify its working.

2. Write a VHDL code for a Full –Adder ,Simulate and verify its working.

3. Write a VHDL code for a 8:1 multiplexer ,Simulate and verify its working.

4. Write a VHDL code for a JK flip flop ,Simulate and verify its working.

Text Book:

1. **Digital Logic and Computer Design:** M. Morris Mano, Pearson (2018), new print 1st edition

Reference Books:

1. **The Elements of Computing System – Building the Modern Computer from First Principles:** Noam Nisan, Shimon Schocken, The MIT Press (2005).
2. **Digital Principles and Applications:** Donald P Leach, Albert Paul Malvino Goutam Saha, TMH, 6th Edition, 2006**.**

3. **Fundamentals of Digital Logic with Verilog Design:** Stephen Brown, ZvonkoVranesic, TMH, 2006
4. **Fundamentals of Logic Design: Charles** H. Roth, Jr., Thomson, 5th Edition, 2004

Online Resources:

1. **NPTEL:** NPTEL: Switching Circuits and logic design https://onlinecourses.nptel.ac.in/noc21_cs64
2. Coursera: Digital Systems - https://www.coursera.org /learn/digital-systems

Course Code: 21BG3C03 | **Course: Biology for Engineers**
Credits: 2 | **L:T:P 2:0:0**
SEE: 50 Marks | **CIE: 50 Marks**
SEE Hours: 2 Hrs | **Max. Marks: 100**

Prerequisites if any	None
Learning objectives	1. To convey that Biology is as important a scientific discipline as Mathematics, Physics and Chemistry 2. Impart knowledge on basic techniques of bioinformatics and analysis of biological data using computational methods

Course Outcomes:

On the successful completion of the course, the student will be able to

Course Outcomes		Bloom's level
CO1	Explain the fundamentals of Life, Evolution, Biomolecules, Cell Biology	Remember Understand
CO2	Explain the importance and applications of bioinformatics	Remember Understand
CO3	Explain the applications of genetics in various field of life sciences	Remember Understand

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3	-	-	-	-	2	2	-	-	-	-	-		-	-
CO2	3	-	-	-	-	2	2	-	-	-	-	-		-	-
CO3	3	-	-	-	-	2	2	-	-	-	-	-		-	-

Strong: 3 Medium: 2 Low: 1

Course Content

	Module - 1 **Introduction to biology for engineers**	**No. of Lecture Hours**	**No. of Tutorial Hours**
1.1	Introduction, Origin of Life & Evolution	2	-
1.2	Biomolecules: The structure, function & significance of Carbohydrates, Proteins, Nucleic acids, & Lipids	2	-
1.3	Cell Structure, types, and organelles. Cell Division	2	-
1.4	Central dogma of molecular biology: DNA replication, transcription, and translation.	2	-
	Module - 2: Computational biology		
2.1	Introduction to bioinformatics and data generation. Tools (FASTA, BLAST), Data base (GENBANK, PubMed, PDB), Data generation (Through Genome sequencing, Protein sequencing)	4	-
2.2	Overview of bioinformatics applications	2	-
2.3	Biosensors: Types and applications	2	-
	Module - 3: Engineering biological systems and its applications		
3.1	Introduction to genetics: Mendelian Genetics	2	-

3.2	Genetic engineering, Methods in genetic engineering, and PCR, ELISA, and its applications,	3	-
3.3	Fundamentals of biomechanics: Neural networks - Stem cell and tissue engineering.	3	-
3.4	Artificial intelligence in biology and plant factory	1	-
Total No. of Lecture Hours		25	
Total No. of Tutorial Hours			0

Text Books:

1. Biology for Engineers by G. K. Suraishkumar; Oxford University Press, 2019, First Edition
2. Dr. Sohini Singh and Dr. Tanu Allen, “Biology for Engineers”, Vayu Education Of India, New Delhi, 2014.
3. Bioinformatics: with fundamentals of genomics and proteomics – Shubha Gopal, et.al., Mc Graw Hill.

Reference Books:

1. Developing Bio informatics computer skills – O’Reilly, CBS. Evolutionary Bioinformatics – Forsdyke, Springer

Online Resources:

1. www.bio12.com/ch3/RaycroftNotes.pdf
2. www.engineering.uiowa.edu/bme050/cvb-solids.pdf
3. www.biologyjunction.com/mendelian_genetics.html

Course: Additional Mathematics-I

Course Code: 21MA3N01 **L:T:P (2:2:0)**

CIE: 50 Marks **Total: 28 Hrs**

Credits: 0 **Max. Marks:50**

Prerequisites if any	Basic concepts of Differentiation and Integration.
Learning objectives	I.To facilitate the students with a strong foundation of differential and integral calculus. **II**. To emphasize the need and importance of Numerical methods to solve engineering problems

Course Outcomes:

On the successful completion of the course, the student will be able to

COs		**Bloom's level**	**Expected attainment level %**
CO1	Expand any differentiable function into power series & compute partial derivatives.	Apply	60
CO2	Compute integrals using appropriate methods and evaluate multiple integrals.	Apply	60
CO3	Estimate a real root of the given equation and apply appropriate interpolation formulae for equal and unequal arguments.	Apply	60

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	S	M										L			
CO2	S	M										L			
CO3	S	M										L			

S – Strong (3) M – Medium(2) L – Low (1)

Course Structure

	Module – 1: Differential Caluculus	**No. of Lecture Hours**	**No. of Tutorial Hours**
1.1	Successive differentiation, Maclaurin's series expansion	2	2
1.2	Taylor's series expansion	1	1
1.3	Partial differentiation,Total derivative	2	2
1.4	Derivative of composite function	2	2
1.5	Maxima,minima for function of two variable	2	2
	Module – 2: Integral Calculus		
2.1	n of definite integrals by the method of substitution	2	2
2.2	Integration by parts, Bernoulli's rule of integration	2	2
2.3	Multiple integrals-Evaluation of double integrals.	1	1
	Module – 3: Numerical Methods		
3.1	Numerical solutions of algebraic and transcendental equations- Newton-Raphson method	2	2
3.2	Finite differences- Forward differences, Backward differences	2	2
3.3	Newton's Forward and Backward Interpolation Formula.	1	1
3.4	Interpolation for unequal intervals-Newton's Divided difference formula	1	1
	Total No. of Lecture Hours	20	
	Total No. of Tutorial Hours		20

Textbook:

1. Higher Engineering Mathematics – B.S. Grewal, 42nd edition, Khanna Publications.

References

1. Advanced Engineering Mathematics – H. K. Dass, Chand Publications.
2. Higher Engineering Mathematics – B. V. Ramana, Tata McGraw-Hill Publications.

Course: Mastering Office

Course Code: 21CS3A01

Credits: 1 **L:T:P – 1:0:0**

CIE: 50 Marks **SEE: 0 Marks**

Prerequisites if any	**NIL**
Learning objectives	1. To attain the knowledge of word processor, spreadsheets and PowerPoint presentation.

<u>Course Outcomes:</u>

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Apply the features of word processing software to create word documents.	Apply (L3)
CO2	Create spreadsheets involving formulas, charts and macros.	Apply (L3)
CO3	Design presentations using templates, graphics and other available options.	Apply (L3)

Mapping with POs and PSOs

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1										3	2	2			
CO2	2			2	2					3	2	2		1	
CO3			1		2					3	2	2		1	

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

No.	Module	No. of Lecture Hours	No. of Tutorial Hours
	Module -1		
1.1	**MS Word** -Working with Files, Text – Formatting, Moving, copying and pasting text.	1	0
1.2	Styles – Lists – Bulleted and numbered lists, Nested lists, Formatting lists. Table Manipulations.	1	0
1.3	Graphics – Adding clip Art, add an image from a file, editing graphics, Page formatting - Header and footers, page numbers.	1	0
1.4	Protect the Document, Mail Merge, Macros – Creating & Saving web pages, Hyperlinks.	1	0
	Module – 2		
2.1	**MS-Excel-** Modifying a Worksheet – Moving through cells, adding worksheets, rows and columns, Resizing rows and columns, selecting cells, Moving and copying cells, freezing panes	1	0
2.2	Macros – recording and running, Linking worksheets	2	0
2.3	Sorting and Filling, Alternating text and numbers with Auto fill, Auto filling functions.	2	0
2.4	Graphics – Adding clip art, add an image from a file, Charts – Using chart Wizard, Copy a chart to Microsoft Word.	2	0
2.4	**MS-Power Point** -Create a Presentation from a template- Working with Slides – Insert a new slide, applying a design template.	1	0
2.5	Changing slide layouts – Resizing a text box, Text box properties, delete a text box	1	0
2.6	Video and Audio effects, Color Schemes & Backgrounds	1	0
2.7	Adding clip art, adding an image from a file, Save as a web page.	1	0
	Total No. of Lecture Hours	**15**	
		Total No. of Tutorial Hours	**0**

Textbooks:

1. Microsoft Office 365 All-In-One for Beginners & Power Users, Tech Demystified, ISBN: B09CVBXKC9, 2021

Online Resources:

1. http://office.microsoft.com/en-us/training/CR010047968.aspx

Course: Data Visualization

Course Code: 21CS3A02

Credits: 1 **L:T:P – 1:0:0**

CIE: 50 Marks **SEE: 0 Marks**

Prerequisites if any	**NIL**
Learning objectives	1. To understand the basics of data visualization. 2. To apply visualization techniques over structured and unstructured data.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand visualization, its importance and its principles.	Understand (L2)
CO2	Apply visualization techniques over structured and unstructured data.	Apply (L3)

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1					3					2	2	2		
CO2	2		2		3					3	2	2	2	2

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

<u>Course Structure</u>

No.	**Module**	**No. of Lecture Hours**	**No. of Tutorial Hours**
	Module -1		
1.1	**Introduction to Visualization:** What Is Data Visualization? Why Do We Have to Visualize Data?	1	0
1.2	How Do We Visualize?	1	0

1.3	Seven Stages of Visualizing Data, Usage of Visualization	1	0
1.4	Types of Charts and Common Chart Selection Questions	1	0
1.5	**Visualization Best Practices:**Importance of Data Visualization	1	0
1.6	Data Types	1	0
1.7	Effectiveness of Visual Encodings, Color	1	0
1.8	Edward Tufte’s Design Principles, Can Chart Junk be Useful?	1	0
	Module – 2		
2.1	**Visualization of Structured Data:** Introduction, Exploratory Analysis	1	0
2.2	Modelling, Visualization during Deployment, Business Operation Dashboard	2	0
2.3	**Visualization of Unstructured Data:** Introduction, Importance of Text Data Visualization, Challenges of Text Data Visualization	1	0
2.4	Various Forms of Text Data	1	0
2.5	Text Data Pre-processing Pipeline, Visualizing Text Data, Visualizing Conversations	2	0
	Total No. of Lecture Hours	**15**	
		Total No. of Tutorial Hours	**0**

Textbooks:

1. Data Visualization: Storytelling Using Data, Sharada Sringeswara; Purvi Tiwari; U. Dinesh Kumar, Wiley India, ISBN: 9789354643132, 2022.

Online Resources:

1. https://www.tableau.com/learn/articles/data-visualization
2. https://analyticsindiamag.com/top-8-open-source-data-visualization-tools/
3. https://www.youtube.com/watch?v=loYuxWSsLNc

Course: Problem Solving and Critical Thinking

Course Code: 21CS3A03

Credits: 1 **L:T:P – 1:0:0**

CIE: 50 Marks **SEE: 0 Marks**

Prerequisites if any	**NIL**
Learning objectives	1. To understand problem solving and critical thinking. 2. To apply problem solving and critical thinking skills to real-world scenarios.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Understand critical thinking, tools and its framework.	Understand (L2)
CO2	Apply different thinking techniques to solve problems.	Apply (L3)

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1					2							2		1	2
CO2	2	1	1	1								2		1	2

Mapping Strength: 3 – Strong 2 – Medium 1 – Low

<u>Course Structure</u>

No.	**Module**	**No. of Lecture Hours**	**No. of Tutorial Hours**
	Module 1		
1.1	**Introduction and the Framework for Critical Thinking**: What Is CriticalThinking?	1	0
1.2	When to use Critical Thinking	1	0
1.3	The Framework and Tools	1	0

	Module – 2		
2.1	**Clarity:** Empty your bucket, Inspection	1	0
2.2	Why? So What? Need	2	0
2.3	Anticipatory thinking, What else?	2	0
2.4	The Ingredient Diagram	1	0
2.5	**Conclusions and Innovation:** Outside-the-Box Thinking	2	0
2.6	Abductive Thinking	2	0
2.7	Impossible Thinking	2	0
	Total No. of Lecture Hours	**15**	
	Total No. of Tutorial Hours		**0**

Textbooks:

1. Think smarter - Critical Thinking to Improve Problem-Solving and Decision-Making Skills, Michael Kallet, Wiley, 2014.

Online Resources:

1. https://www.utc.edu/academic-affairs/walker-center-for-teaching-and-learning/faculty-support-and-resources/pedagogical-strategies-and-techniques/ct-ps

IV SEMESTER

Statistics and Probability

(CIV/MECH/IP/CS/IS)

Course Code: 21MA4C01

Credits:3 **L:T:P (3:0:0)**

SEE: 50% Marks **CIE: 50% Marks**

SEE Hours: 3 hours **Max. Marks:100**

Prerequisites if any	Basic knowledge of set theory, counting techniques and calculus
Learning objectives	**I**To understand the concepts of random variables, probability distributions, Markov chains and queuing systems **II**To extract maximum information about the population by examining the samples of the population

Course Outcomes:

On the successful completion of the course, the student will be able to

COs		**Bloom's level**	**Expected attainment level %**
CO1	To solve problems associated with discrete and continuous probability distributions of one or more variables	Apply	60
CO2	To solve problems associated with Markov chains and queuing systems	Apply	60
CO3	To study skewness and kurtosis for a given data and use the method of least squares to fit curves for a given data and solve problems on correlation and regression	Apply	60
CO4	To test the hypothesis for sampling distributions	Apply	60

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	S	M										L			
CO2	S	M										L			
CO3	S	M										L			
CO4	S	M										L			

S – Strong (3) **M – Medium(2)** **L – Low (1)**

Course Structure

	Module – 1:Basic Probability and Discrete Probability Distribution	**No. of Lecture Hours**	**No. of Tutorial Hours**
1.1	Probabilistic models, conditional probability, Mutually exclusive, independence events	1	Nil
1.2	Total probability, Baye's theorem, Counting	1	Nil
1.3	Random variables-probability mass functions	1	Nil
1.4	Mathematical expectations, moment generating functions	2	Nil
1.5	Discrete distributions: Binomial	2	Nil
1.6	Poisson distributions	1	Nil
	Module-2: Continuous and Joint Probability Distribution		
2.1	Continuous random variables & PDF's, Cumulative distribution functions	3	Nil
2.2	Normal distributions,Central limit theorem	2	Nil
2.3	Two dimensional Random variables: Joint probability distributions	3	Nil
	Module-3: Markov Chains and Queuing Theory		
3.1	Markov chains – probability vector	2	Nil
3.2	Stochastic matrix, transition probability matrix	3	Nil
3.3	Concept of queuing – M/M/1 queuing system	2	Nil
	Module-4: Statistics		
4.1	Moments, skewness from third moment	1	Nil
4.2	Measure of skewness by Karl Pearson's, Kurtosis	2	Nil
4.3	Curve fitting- Straight line and Exponential curves-y=ab^x	2	Nil
4.4	Correlation, Regression	2	Nil
4.5	Multiple Correlation and Regression.	2	Nil
	Module-5: Sampling Theory		
5.1	Population and Sampling, Sampling distributions, Statistical estimation	2	Nil
5.2	Statistical Decisions: Type I and Type II errors	2	Nil
5.3	Level of Significance, One-tailed tests, ***t***-distribution	1	Nil
5.4	Chi-Square distribution.	3	Nil
	Total No. of Lecture Hours	40	
		Total No. of Tutorial Hours	Nil

Text Books:

1. Higher Engineering Mathematics – B.S. Grewal, 42nd edition, Khanna Publications.

2. Introductory probability and statistical applications, Paul L Meyer, Oxford and IBH publishing co.pvt. Ltd, New Delhi

References:

1. Probability and Stochastic processes, Roy D Yates, David J Goodman, Wiley publication, 3rd edition.

2. Advanced engineering mathematics, Erwin Kreyszig, 8th edition, Wiley publication.

Course: Design and Analysis of Algorithms

Course Code: 21CS4C01

Credits: 4	**L:T:P 3:0:2**
CIE: 50% Marks	**SEE: 50% Marks**
SEE Hours: 3 hours	**Max. Marks: 100**

Prerequisites if any	**Mathematics of recurrence relations, Data structures**
Learning objectives	1. To learn different algorithm design strategies. 2. To design algorithms using various design techniques 3. To analyze different algorithms with respect to their time complexities

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Analyze the algorithms using space and time complexity; compare their order of growth using asymptotic notations.	L2, L3
CO2	Design algorithms using divide and conquer method and analyze the different algorithms.	L2, L3
CO3	Apply the greedy strategy to solve different problem statements.	L3
CO4	Illustrate the use of dynamic programming method to solve the different problems.	L2,L3
CO5	Apply the Backtracking and Branch and Bound methods to solve few classic problems.	L2, L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	2	1									1		3	2
CO2	3	2	3									1		3	2
CO3	2	2	3									1		3	2
CO4	2	2	3									1		3	2
CO5	2	2	3									1		3	2

Mapping strength : 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No. of Lecture Hours	No. of Tutorial Hours
	Module – 1		
1.1	**Algorithm Performance Analysis:** Introduction,	1	0
1.2	Space and Time Complexities	2	0
1.3	Asymptotic growth rate and notations.	2	0
1.4	Case studies for performance analysis : Rank sort	1	0
1.5	Case studies for performance analysis : Insertion sort	2	0
	Module – 2		
2.1	**Divide and Conquer:** The method,	1	0
2.2	Recurrence relations	2	0
2.3	The Master theorem,	1	0
2.4	Applications: Binary Search, Merge sort,	2	0
2.5	Quick sort	1	0
2.6	Matrix multiplication (Strassen’s method)	1	0
	Module – 3		
3.1	**The Greedy Method:** Elements of the Greedy Strategy	1	0
3.2	Applications :Bipartite Cover-Max cover problem	1	0
3.3	Topological Sorting	1	0
3.4	Single Source Shortest paths**:** Dijkstra‟s Algorithm	2	0
3.5	Minimum-Cost Spanning Tree Algorithms : Kruskal algorithm	2	0
3.6	Prim’s algorithm	1	0

	Module – 4		
4.1	**Dynamic Programming:** The Elements of Dynamic Programming,	1	0
4.2	Applications : Binomial Coefficient,	1	0
4.3	0/1 Knapsack Problem,	2	0
4.4	Longest Common Subsequence,	2	0
4.5	All Pairs Shortest Path	2	0
	Module – 5		
5.1	**Back Tracking:** The Method	1	0
5.2	Applications :n-Queens Problem	2	0
5.3	Sum of subsets Problem	2	0
5.4	**Branch and Bound Method**: The Method	1	0
5.5	Application : Travelling Salesman problem.	2	0
	Total No. of Lecture Hours	**40**	
		Total No. of Tutorial Hours	**0**

Text Books:

1. **Introduction to Algorithms,** Cormen Et Al. PHI, 3rd Edition.
2. **Introduction to Design & Analysis of Algorithms,** Anany Levitin, Pearson, 3rd Edition.

Reference Books:

1. **Computer Algorithms – Introduction to Design and Analysis,** Sara Baase, Allen VanGelder, Pearson Education, 3rd Edition.
2. **The Design and Analysis of Computer Algorithms,** Alfred V Aho, John E Hopcroft, Jeffrey D Ullman, Pearson Education.
3. **Algorithms:** S.Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, McGraw-HillScience/ Engineering/ Math.

Online Resources:

1. **https://nptel.ac.in/courses/106/101/106101060/**
2. **https://www.coursera.org/learn/analysis-of-algorithms**

Integrated Lab Component: Design and Analysis of Algorithms

Sl. No	COs	TITLE OF PROGRAMS
1.	CO1	Sort a given set of elements using Rank sort and Repeat the experiment for different values of n, the number of elements in the list to be sorted. Analyze the time complexity
2.	CO1	Sort a given set of elements using Insertion sort method and Repeat the experiment for different values of n, the number of elements in the list to be sorted. Analyze the time complexity.
3.	CO1	Sort a given set of elements using Selection sort and hence find the time required to sort elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
4.	CO2	Sort a given set of elements using Merge sort method and determine the time taken to sort the elements. Repeat the experiment for different values of n.Analyze the time complexity.
5.	CO2	Sort a given set of elements using Quick sort method. Repeat the experiment for different values of n.Analyse the time complexity.
6.	CO3	Obtain the Topological ordering of vertices in a given digraph.
7.	CO3	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
8.	CO3	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm
9.	CO3	Find Minimum Cost Spanning Tree of a given undirected graph using Prims algorithm.
10.	CO4	Implement Transitive closure using Warshall’s algorithm.
11.	CO4	Implement All Pair Shortest paths problem using Floyd’s algorithm**.**
12.	CO4	Implement 0/1 Knapsack problem using Dynamic programming.
13.	CO4	Find the Binomial Co-efficient using Dynamic programming.
14.	CO5	Open ended assignments on algorithms based on Backtracking Strategy
15.	CO5	Open ended assignments on algorithms based on Branch and Bound Strategy

Course: Formal Languages and Automata Theory

Course Code: 21CS4C02

Credits: 4 **L:T:P - 3:2:0**

CIE: 50% Marks **SEE: 50% Marks**

SEE Hours: 3 hours **Max. Marks: 100**

Prerequisites if any	**Discrete Mathematics and Logic Design**
Learning objectives	1. To design different types of Finite Automata 2. To understand the relationship between FA, RE, PDA and CFGs. 3. To convert a Grammar to a normal form 4. Acquire knowledge to design basic TMs.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Interpret the fundamentals of formal languages and Design different types of finite automata.	L2, L3
CO2	Illustrate the relationship between finite automata and Regular Expressions.	L3
CO3	Demonstrate the equivalence between FA and Context Free Grammar and design CFG for specified problems.	L3
CO4	Design a PDA and evaluate the equivalence with Context Free Grammar	L3
CO5	Illustrate the methods of normalization on CFGs	L3
CO6	demonstrate the abstract model of computing machine through Turing Machines	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	2	2	3											2	2
CO2	2	2	3		1									2	2
CO3	2	2	3		2									2	2
CO4	2	2	3											2	2
CO5	2	2	3		2									2	2
CO6	2	2	3		2									2	2

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No. of Lecture Hours	No. of Tutorial Hours
	Module – 1		
1.1	**Introduction To Finite Automata:** Introduction to Finite Automata, The central concepts of Automata theory	1	0
1.2	Deterministic finite automata (Definition and problems)	4	0
1.3	Nondeterministic finite automata (Definition and problems)	2	0
1.4	Conversion of NFA to DFA	1	0
1.5 **(T)**	Problems on DFA and NFA	0	2
	Module – 2		
2.1	**Nondeterministic finite with Epsilon-transitions**: Introduction to –Epsilon-NFA	2	0
2.2	Epsilon-NFA to DFA	2	0
2.3	Equivalence of automata	1	0
2.4	Minimization of automata	1	0
2.5 **(T)**	Problems of Epsilon NFA , equivalence and minimization of FA	0	2
	Module – 3		
3.1	**Regular expressions:** Introduction and problems	2	0

3.2	**Finite Automata and Regular Expressions:** FA to RE	1	0
3.3	RE to FA	1	0
3.4	Proving languages not to be regular languages (Pumping Lemma);	1	0
3.5 **(T)**	Problems of RE and Conversion of RE to FA	0	2
	Module – 4		
4.1	**Context-Free Grammars and Languages:** Context –free grammars;	3	0
4.2	Parse trees; Ambiguity in grammars and Languages.	2	0
4.3	**Pushdown Automata:** Definition of Pushdown automata	2	0
4.4	The languages of PDA	2	0
4.5 **(T)**	Problems of Design of CFG and PDA	0	2
	Module – 5		
5.1	Deterministic Pushdown Automata	2	0
5.2	**Equivalence of PDAs and CFGs:** Grammar to PDA	1	0
5.3	PDA to Grammar	1	0
5.4	Eliminating useless symbols, Epsilon productions and Unit productions from a Grammar	3	0
5.5 **(T)**	Problems on DPDA Conversion of CFG to PDA and PDA to CFG	0	1
	Module – 6		
6.1	**Normal forms for CFGs:** Chomsky Normal Form (CNF)	2	0
6.2	**Introduction to Turing Machine:** Definition of Turing Machine	2	0
6.3	Nondeterministic Turing Machines	1	0
6.4 **(T)**	Problems on CNF and Turing Machine	0	1
	Total No. of Lecture Hours	***40***	
		Total No. of Tutorial Hours	***10***

Text Books:

1. **Introduction to Automata Theory, Languages and Computation:** John E.Hopcroft, Rajeev Motwani, Jeffrey D.Ullman:, Pearson education, 3 rd Edition, 2008

Reference Books:

1. **Introduction to Languages and Automata Theory**, John C Martin, TataMcGraw-Hill, 3rd Edition, 2007.
2. **Introduction to the Theory of Computation:** (2nd edition). Michael J.Sipser, Thomson (Coursetechnology),2006

Online Resources :

https://nptel.ac.in/courses/106/105/106105196/

https://www.edx.org/course/automata-theory?index=product&queryID=ec87b28bffaf63ed20d2ed479aeb5f7e&position=1

Course: Object Oriented Programming

Course Code: 21CS4C03
Credits: 3 **L:T:P - 3:0:0**
CIE: 50% Marks **SEE: 50% Marks**
SEE Hours: 3 **Max. Marks: 100**

Prerequisites if any	**C Programming**
Learning objectives	1. To provide the knowledge of fundamental principles of Object Oriented Programming. 2. To introduce the concepts of C++ language such as class, object, inheritance, overloading, virtual functions, STL etc.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Explain principles of Object Oriented Programming using class and objects.	L2
CO2	Illustrate the concepts of function and operator overloading.	L3
CO3	Demonstrate reusability using inheritance and virtual functions.	L3
CO4	Apply function template and class template for C++ programs.	L3
CO5	Illustrate STL and Exception Handling using real world problems.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3		2	1	3				1	1	1	2			3
CO2	3		2	1	3				1	1	1	2			3
CO3	3		2	1	3				1	1	1	2			3
CO4	3		2	1	3				1	1	1	2			3
CO5	3		2	1	3				1	1	1	2			3

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	**Modules**	**No. of Lecture Hours**	**No. of Tutorial Hours**
	Module – 1	**9**	
1.1	**Classes and Objects:** An overview of C++,classes and objects, Relationship of Structure, Union and Class in C++	1	0
1.2	Friend Functions, Friend Classes, Inline Functions- Defining Inline Functions Within a Class	1	0
1.3	Constructors and Destructors, Parameterized Constructors-Constructors with One Parameter: A Special Case, copy constructor.	1	0
1.4	Static Class Members- Static Data Members and Static Member Functions, When Constructors and Destructors Are Executed, Passing Objects to Functions, Returning Objects.	2	0
1.5	**Arrays, pointers, References, and the Dynamic Allocation Operators:** ‘this’ Pointer, References – Reference Parameters, Passing References to Objects and Returning References.,	2	0
1.6	C++‘s Dynamic Allocation Operators- Initializing Allocated Memory, Allocating Arrays and Allocating Objects	2	0
	Module – 2	**8**	
2.1	**Function Overloading, Copy Constructor, and Default Arguments:** Function Overloading	1	0
2.2	Copy Constructor, Default Function Arguments, Default Argument vs. Overloading.	2	0

2.3	**Operator Overloading:** Creating a Member Operator Function- Creating Prefix and Postfix Forms of the Increment and Decrement Operators,	2	0
2.4	Operator Overloading Using a Friend Function – Using a Friend to Overload ++ or --, Friend operator Functions Add Flexibility, Overloading << and >>.	3	0
	Module – 3	**9**	
3.1	**Inheritance:** Base-Class Access Control, Inheritance and protected Members- Protected Base-Class Inheritance, Inheritance Multiple Base Classes	3	0
3.2	Constructors, Destructors, and Inheritance- When Constructors and Destructors Are Executed, Passing Parameters to Base-Class Constructors. Granting Access, Virtual Base Classes.	2	0
3.3	**Virtual Functions and Polymorphism:** Virtual Functions- Calling a Virtual Function Through a Base- Class Reference, The Virtual Attribute vs. Inherited,	2	0
3.4	Virtual Functions Are Hierarchical, Pure Virtual Functions- Abstract Classes, Using Virtual Functions, Early vs. Late Binding	2	0
	Module – 4	**7**	
4.1	**Templates:** Generic Functions- A Function with Two Generic Types, Explicitly Overloading a Generic Function,	1	0
4.2	Overloading a Function Template, Using Standard Parameters with Template Functions, Generic Function Restrictions.	2	0
4.3	**Applying Generic Function:** A generic Sort.	2	0
4.4	**Applying Generic Classes:** An Example with Two Generic Data Types, A Generic Array Class, Using Non-Type Arguments with Generic classes, Using Default Arguments with Template Classes.	2	0
	Module – 5	**7**	
5.1	**Exception Handling:** Fundamentals, Handling derived class Exception	2	0
5.2	Exception Handling options.	2	0
5.3	**Introducing the Standard Template Library:** An Overview of the STL, Container Classes,	1	0
5.4	General Theory of Operation, Vector container class, Algorithms.	2	0
	Total No. of Lecture Hours	**40**	
		Total No. of Tutorial Hours	**0**

Text Books:

1. **C++ The Complete Reference,** Herbert Schildt, TMH, McGraw-Hill, 4th Edition, 2017.

Reference Books:

1. **The C++ programming language,** Bjarnestroustrup , Pearson Education, 3rd Edition, 2013.
2. **C++ Primer**, Stanley B.Lippman and JoseeLajore, Addison Wesley, 3rd Edition, 2014.
3. **WEBLINK:** http://www.cplusplus.com/

Online Resources:

1. NPTEL: Programming in modern C++ https://nptel.ac.in/courses/noc22_cs43
2. Coursera: Programming in C++- https://www.coursera.org/specializations/hands-on-cpp

Course: Software Engineering

Course Code: 21CS4C04

Credits: 3 **L:T:P 3:0:0**

CIE: 50% Marks **SEE: 50% Marks**

SEE Hours: 3 Hours **Max. Marks: 100**

Prerequisites if any	**NIL**
Learning objectives	1. Learn the fundamentals of software engineering process and process models 2. Learn to use appropriate analysis and modeling techniques for building a software systems for real world problems 3. Learn to validate the software systems using testing strategies 4. Use suitable software projectestimation model for developing software

<u>Course Outcomes:</u>

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Describe the fundamentals of Software Engineering Process and Process Models	L2 Understand
CO2	Discuss requirement engineering tasks.	L2 Understand
CO3	Prepare quality software system using design principles.	L3 Apply
CO4	Use software testing techniques to perform system validations.	L3 Apply
CO5	Apply an effective software project estimation model for developing software product.	L3 Apply

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3		2						2	2		2			
CO2	3	2			2						2	2		2	
CO3			3		2				2	2		2		3	
CO4	3			2								2			2
CO5	3	3	3	2		2		2	2	2	2	2			3

Mapping strength : 3 – Strong 2 – Medium 1 – Low

Course Structure

Nos	Modules	No. of Lecture Hours	No. of Tutorial Hours
	Module – 1		
1.1	**Introduction** to Software engineering	2	0
1.2	**The Software Process**: Software Engineering, A Layered Technology, A Process Frame Work, Capability Maturity Model Integration	3	0
1.3	**Process Models**: Incremental Process Models, Evolutionary Process Models	3	0
	Module – 2		
2.1	**Agile View of Process**: Agility, Agile Process, Agile Process Model	3	0
2.2	**Requirement Engineering:** Requirement Engineering Tasks, Initiating Requirement Engineering Process, Developing USE-CASE	4	0
	Module – 3		
3.1	**Building The Analysis Model**: Requirement Analysis, Analysis Modeling Approach, Data Modeling concept, Scenario Based Modeling, Flow Based Modeling, and Behavioral Modeling	4	0
3.2	**Design Engineering:** Design Process and Design Quality, Design Concepts	4	0
3.3	**Creating an Architectural Design**: Software Architecture, Data Design, Architectural Styles and Patterns	3	0
	Module – 4		

4.1	**Testing Strategies**: A Strategic Approach to Software Testing, Test Strategies for Conventional Software, validation testing	3	0
4.2	**Testing Tactics**: Software Testing Fundamentals, Black Box & White Box Testing, Basis Path Testing, Black Box Testing	4	0
	Module – 5		
5.1	**Project Management**: Project Management Spectrum, People, Product, Process, Project	3	0
5.2	**Software Project Estimation**: Decomposition Techniques, Empirical Estimation Models	3	0
5.3	**Report writing**	1	0
Total No. of Lecture Hours		**40**	
Total No. of Tutorial Hours			**0**

Text Books:

1. Software Engineering: A Practitioners Approach – Roger S. Pressman, 7th Edition, McGraw-Hill 2010

Reference Books:

1. Software Engineering: Ian Somerville, 10 th Edition, Pearson Education, 2016.

2. Software Engineering Theory and Practice : Shari Lawrence Pfleeger, Joanne M. Atlee, 3rd Edition, Pearson Education, 2006.

3. Software Engineering Principles and Practice: Waman S Jawadekar,Tata McGraw Hill, 2004

Online Resources:

1. https://www.digimat.in/nptel/courses/video/106101061/L01.html
2. https://www.digimat.in/nptel/courses/video/106105182/L01.html
3. https://www.coursera.org/learn/software-processes-and-agile-practices

Course: Object Oriented Programming Laboratory

Course Code: 21CS4L01

Credits: 1 **L:T:P 0:0:2**

CIE: 50% Marks **SEE: 50% Marks**

SET Hours: 2 hours **Max. Marks: 25**

Prerequisites if any	**Fundamentals of C programming**
Learning objectives	1. To provide the knowledge of fundamental principles of Object Oriented Programming . 2. To introduce the concepts of C++ language such as class, object, inheritance, overloading, virtual functions, STL etc.

<u>Course Outcomes:</u>

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Demonstrate class and object concepts using C++.	L3
CO2	Develop programs using function overloading and operator overloading.	L3
CO3	Demonstrate reusability of classes using inheritance and virtual functions.	L3
CO4	Construct generic classes using template concepts.	L3
CO5	Implement STL and Exception Handling.	L3

<u>Mapping with POs and PSOs</u>:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	3		2	1	3				1	1	1	2			3
CO2	3		2	1	3				1	1	1	2			3
CO3	3		2	1	3				1	1	1	2			3
CO4	3		2	1	3				1	1	1	2			3
CO5	3		2	1	3				1	1	1	2			3

Mapping strength : 3 – Strong 2 – Medium 1 – Low

List of experiments

1. a. Write a C++ program to read three numbers from the keyboard and display the largest value on the screen.

   b. Write a C++ program to check whether the given number is an Armstrong number or not.(Hint: Armstrong number 153=1^3 + 5^3 + 3^3).

   c. Write a C++ program to find and display Factorial of a number. (Using iterative method or recursion).

   d. Write a C++ program to print ASCII value of a character.

   e. Write a C++ program that will accept an array of numbers and display the number of times the given number occurred in the array.

2. a) Convert time from HH:MM:SS format to seconds using class in C++.

   b) A phone number, such as (044) 234-8900, can be thought of as having three parts: the area code (044), the exchange (234) and the number (8900). Write a program that uses a class to store these three parts of a phone number separately. Call the class phone. Create two class objects of type phone. Initialize one, and have the user input a number for the other one. Display both the numbers.

3. Create two classes DM and DB which store the value of distances. DM stores distance in meters and centimetres and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimetres depending on the object on display.

4.Write a C++ program to create a class Rectangle with data members: length, breadth, area and member functions as:

i) Rectangle&setDimensions(const int&, const int&) – sets the length and breadth of arguments using constant references. Use *this* pointer to return the resized Rectangle by reference.

ii) int computeArea() – to compute and returns area of a rectangle.

iii) int computePerimeter() – to compute and return perimeter of a rectangle.

iv) Two constructors, default constructor to initialize data members to zero and an overloaded constructor as arguments with breadth having a default value.

Define all the member functions outside the class. Create objects of Rectangle type and test all the functions. Delete the objects before program terminates.

5. Write a C++ program to create a class called Complex and implement the following overloading member functions that return a Complex number after performing addition of input args.

- Complex ADD(int a, Complex s) – where a is an integer(real part) and s is a complex number.
- Complex ADD(Complex &s1, Complex &s2)Create objects of *Rectangle* type on heap utilizing both constructors. Use the member functions to modify the rectangle size, compute area, perimeter. Display the area & perimeter of each rectangle. Delete the objects before pgm termination.

6. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators ‘+’ and ‘-’.

i) S1 = S1 + element; where S1 is the object of class STACK and element is an integer to be pushed on the top of stack.

ii) int element = S1--; where S1 is the object of class STACK. ‘—‘operator pops the top element.

Handle the STACK empty and full conditions and also display the contents after every operation by overloading << operator.

7.Write a C++ program to read and print Employee information (name, empID, gender) with Department (deptName, workAssigned) and with Loan information (loanDetails,loanAmt) using hierarchical inheritance.

8. Write a C++ program to design a Student class representing USN and a Test class representing the scores of the student in various subjects and a Sports class representing the score in sports. The Sports and Test classes is inherited by Result class having the functionality to add the scores and display the final result of a student.

9. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5

students from UG and PG. Find the average age for all UG and PG students separately

10.Implement class *Shape* with the following specification*:*

class Shape{

protected:

float area, perimeter; public:

Shape();

virtual void initialize()=0; virtual float
computeArea()=0;

virtual float computePerimeter()=0; virtual
~Shape();

};

Implement 2 classes *Triangle* and *Rectangle* publicly derived from class *Shape*, with suitable data members. Implement all the functions derived from class *shape* in each of the derived classes. Write a C++program to create objects of each of the derived class and assign to the base class (Shape) type pointer/reference. Demonstrate runtime

polymorphism by calling the functions of the derived class objects by using the base class pointer/reference

11.a)Write two function templates in C++ to

i) Sort the numbers

ii) To search a given number

Demonstrate the above functions on array of integers and double.

b) A point on the 2D can be represented by two numbers: an x co-ordinate and a y co-ordinate. The sum of two points can be defined as a new point whose x co- ordinate is the sum of x co-ordinates of both points and same for y co-ordinates. Using

function template, find the third point in C++.

12.a) Write a simple calculator using class template in C++

b) Write a program implementing stack and its operations using template class.

13.Write a C++ program with the following:

a. A function to read two double type numbers from keyboard.

b. A function to calculate the division of these two numbers.

c. A try block to throw an exception when a wrong type of data is keyed in.

d. A try block to detect and throw an exception if the condition ―divide-by-zero occurs

e. Appropriate catch block to handle the exceptions thrown.

14.a)Perform these basic vector operation using Standard Template Library:

i) Find the number of elements in the vector.

ii) Check whether the vector is empty or not.

iii) Insert some elements into the vector.

iv) Remove the element at a particular position.

v) Find the index of a particular element in a vector.

b) Make a vector of random numbers and sort it in descending order using STL and also find its sum.

Text Books:

1. **C++ The Complete Reference,** Herbert Schildt, TMH, McGraw-Hill, 4th Edition, 2017.

Reference Books:

1. **The C++ programming language,** Bjarnestroustrup, Pearson Education, 3rd Edition, 2013.
2. **C++ Primer**, Stanley B.Lippman and JoseeLajore, Addison Wesley, 3rd Edition, 2014.
3. **WEBLINK:** http://www.cplusplus.com/

Online Resources:

1. NPTEL: Programming in modern C++ https://nptel.ac.in/courses/noc22_cs43
2. Coursera: Programming in C++- https://www.coursera.org/specializations/hands-on-cpp

Course: Additional Mathematics-II

Course Code:21MA4N01 **CIE: 50 Marks**

Credits: 0 **L:T:P (2:2:0)**

Prerequisites if any	Matrix Theory and Ordinary Differential Equations.
Learning objectives	I. To develop a strong foundation of Linear Algebra in a comprehensive manner. II. To impart knowledge of various methods of solving first and higher order ordinary differential equations. III. To emphasize the need and importance of Numerical methods to solve engineering problems.

Course Outcomes:

On the successful completion of the course, the student will be able to

COs		**Bloom's level**	**Expected attainment level %**
CO1	Operate elementary transformations on matrices to solve system of linear equations, compute eigenvalues and eigenvectors.	Apply	50
CO2	Solve first and higher order linear differential equations.	Apply	50
CO3	Estimate the values of the derivatives and definite integrals using numerical techniques. Use numerical techniques to solve ordinary differential equation with initial conditions.	Apply	50

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1	S	M										L			
CO2	S	M										L			
CO3	S	M										L			

S – Strong (3) M – Medium(2) L – Low (1)

Course Structure

	Module – 1: Linear Algebra	**No. of Lecture Hours**	**No. of Tutorial Hours**
1.1	Elementary transformations of a matrix. Rank of a matrix by elementary row transformations	2	2
1.2	Consistency of a system of linear algebraic equations, Solution of a system of non homogeneous equations	2	2
1.3	Eigen values and Eigen vectors of a square matrix	2	2
	Module – 2: Differntial Equations		
2.1	Solution of first order and first degree differential equations-Separation of Variables linear, Exact.	3	3
2.2	Solution of higher ordernon-homogeneous differential equations- P.I. for e^{ax}, sinax/cosax,x^n.	3	3
	Module – 3: Numerical Methods		
3.1	Numerical Differentiation- equal intervals.	2	2
3.2	Numerical integration- Trapezoidal rule, Simpson's 1/3rd, Simpson's 3/8th & Weddle's rules	3	3
3.3	Numerical solutions of ODE: Taylor's series method ,Modified Euler's method	2	2
3.4	Runge-Kutta method.	1	1
	Total No. of Lecture Hours	20	
		Total No. of Tutorial Hours	20

Text Book:

1. Higher Engineering Mathematics – B.S. Grewal, 42nd edition, Khanna Publications

References

1. Advanced Engineering Mathematics – H. K. Dass, Chand Publications.
2. Higher Engineering Mathematics – B. V. Ramana, Tata McGraw-Hill Publications.

Course: Professional Communication

Course Code: 21CS4A01

Credits: 1 **L:T:P- 1:0:0**

CIE: 50 Marks **SEE: 0**

Prerequisites if any	No Prerequisites
Learning objectives	1. Enhance the Employability and Career Skills of students 2. Orient the students towards grooming as a professional and develop their confidence to attend interviews successfully

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	**Course Outcomes**	**Bloom's level**
CO1	Develop adequate Soft Skills required for the workplace	L2
CO2	Participate confidently in Group Discussions and in Interviews	L2

Mapping with POs and PSOs:

COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**		**PSO1**	**PSO2**
CO1							2		3	3		2			
CO2							2		3	3		2			

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

No.	Module	No. of Lecture Hours	No. of Tutorial Hours
	Module-1		
1.1	**Introduction to Professional Communication:** Role and Importance of Communication,Principles of Effective Communication,Group Communication, Problems of Group Communication	2	0
1.2	**Meetings:** Types of Meetings, Purpose of Meeting, Committee Meeting, The advantages, and disadvantages in using a committee for decision-making committee	2	0
1.3	**Motivational Theories:** Maslow's Need-Hierarchy Theory, Herzberg's hygiene factor theory of Motivation, Comparison of Herzberg, and Maslow Model	2	0
1.4	**Decision Making:** Important Features of Decision Making, Types of Decisions, Decision Making Process	2	0
	Module – 2		
2.1	**Interview:** Types of Interviews	1	0
2.2	**Attitude**: Definitions of Attitudes, Features of Attitude, Components of Attitude, Functions of Attitudes	3	0
2.3	**Business Etiquette:** Introduction to Business Etiquette	1	0
2.4	**Group Discussion:** Importance of GD, Dos for GD, Don'ts for GD, GD Tips	2	0
Total No. of Lecture Hours		**15**	
Total No. of Tutorial Hours			**0**

Textbook:

1. Professional Communication, Ashutosh Srivastava, Dr. Arindam Chatterjee, 1st Edition 2011.

Reference Books:

1. Butterfield, Jeff Soft Skills for Everyone. Cengage Learning: New Delhi, 2015

Online Resources:

1. https://www.youtube.com/watch?v=UNUilGw5IbI
2. https://www.youtube.com/watch?v=am3--dafLj8

Course: Graphic Design

Course Code: 21CS4A02

Credits: 1 **L:T:P-1:0:0**

CIE: 50 Marks **SEE: 0**

Prerequisites if any	NIL
Learning objectives	1. Define graphic communication and so on 2. Define design as a problem- solving tool for communication.

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Understand the Basics of Graphic Design.	L1
CO2	Utilize relevant applications of tools and technology in the creation, reproduction, and distribution of visual messages.	L2

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1					3										
CO2						3		2		2					

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

No.	Module	No. of Lecture Hours	No. of Tutorial Hours
	Module-1		
1.1	Communication Goals: What Is Graphic Communication? Information	3	0
1.2	Persuasion, Education, Entertainment.	4	0
	Module – 2		
2.1	Free tools for Graphic design, Introduction to Adobe Express tool, adding multiple pages, removing background, image effect, create templates.	4	0

2.2	Introduction to Canva tool, creating banners, info graphics, Logos, Posters, Videos editing.Animation.	4	0
Total No. of Lecture Hours		**15**	
Total No. of Tutorial Hours			**0**

Text Books:

1. **Digital Design for Print and Web**: An Introduction to Theory, Principles, and Techniques by John DiMarco , John Wiley and Sons, 2010.

Online Resources:

2.1 Adobe Introduction:https://www.adobe.com/express/create/social-media-graphic

2.1Introduction to Adobe Express tool: https://www.adobe.com/express/learn/tutorials

2.2 Introduction to Canva tool:https://www.canva.com/create/

https://www.canva.com/designschool/tutorials/getting-started/

Animation:https://www.canva.com/en_in/pro/animator/

Course: Digital Design

Course Code: 21CS4A03

Credits: 1 **L:T:P – 1:0:0**

CIE: 50 Marks **SEE: 0**

Prerequisites if any	NIL
Learning objectives	1. To gain knowledge about Digital Design, its elements, and Design Techniques

Course Outcomes:

On successful completion of the course, the student will be able to:

COs	Course Outcomes	Bloom's level
CO1	Understand the digital design process and its elements.	L2
CO2	Apply graphic techniques to create digital design depending on products.	L3

Mapping with POs and PSOs:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2
CO1								1	1	3		2			
CO2								1	1	3		3			

Mapping strength: 3 – Strong 2 – Medium 1 – Low

Course Structure

No.	Module	No. of Lecture Hours	No. of Tutorial Hours
	Module-1		
1.1	Design: Design as a Problem-Solving Tool - What Is Design?	1	0
1.2	Convergent versus Divergent, Two- Three- and Four-Dimensional Design	1	0
1.3	Design Devices: Space, Format, and Structure	1	0
1.4	Design Elements: The Visual Pieces: Line, Shape, Texture, Value, Color and Color Wheel	1	0

1.5	Typography Considerations and Design Principles	2	0
1.6	Seven-Step Design Process	1	0
1.7	Digital Design Project Types: Annual Reports and Business Reports - Announcements, Invitations, and Postcards -Advertising -Books - Brochures and Collateral - Calendars - Catalogs - Direct Mail and Direct Response - Environmental Graphics - Internet Design - Letterhead and Stationery (Corporate Identity) - Logos -Newsletters - Packaging -POP, Signs, and Billboards -Posters -Publication Design	2	0
	Module – 2		
2.1	Techniques - Raster Graphics Techniques: Document Setup, Layers and Compositing	1	0
2.2	Selections	1	0
2.3	Masks, Color	1	0
2.4	Shape Basics, Retouching Basics	1	0
2.5	Type in Digital Imaging Applications, Bitmap Filters, Layer Effects	1	0
2.6	Final Output for Print, Final Output for Web, Design Assignment	1	0
	Total No. of Lecture Hours	**15**	
		Total No. of Tutorial Hours	**0**

Text Books:

1. John DiMarco, **"Digital Design For Print and Web - An introduction to theory principles and design",** 2010**,** John Wiley and Sons.

Online Resources:

1. https://www.canva.com/learn/print-vs-web/
2. https://geekflare.com/open-source-designer-tools/