Dayananda Sagar University

School of Engineering

Innovation Campus, Kudlu Gate, Hosur Road, Bengaluru

Department of CSE (Data Science)

SCHEME AND SYLLABUS

B.Tech. PROGRAMME– 2022 BATCH

Academic Year 2023-24

(3rd semester & 8th semester)

Definitions / Descriptions

Definition of Credit:	
1 Hour Lecture (L) Per Week	01 Credit
1 Hour Tutorial (T) Per Week	01 Credit
1 Hour Practical (P) Per Week	0.5 Credit
1 Hour Project (J) Per Week	0.5 Credit

Course code and Definition:	
BSC	Basic Science Courses
ESC	Engineering Science Courses
HSMC	Humanities and Social Sciences including Management Courses
IPCC	Integrated Professional Core Course
PCC	Professional Core Courses
PEC	Professional Elective Courses
OEC	Open Elective Courses
SEC	Skill Enhancement Courses
UHV	Universal Human Value Course
PROJ	Project Work
INT	Internship

Implementation of National Education Policy (NEP) 2020 for the B.Tech students of Batch 2022-2026

The implementation of Curriculum follows NEP 2020 and addresses the following features and categories of courses:

1. Student Centric flexible curriculum.
2. Interdisciplinary Courses,
3. Multi-disciplinary Courses,
4. Ability Enhancement Courses,
5. Skill Enhancement Courses,
6. Value Added Courses,
7. Product Design and Development,
8. Internship (Rural Internship, Industry Internship, Research/Development Internship), and
9. Multiple Exit and Multiple Entry
    - Certificate in Engineering after completion of first year.
    - Diploma in Engineering after completion of second year.
    - Advanced Diploma in Engineering after completion of third year.
    - Degree in Engineering after completion of fourth year.

Dayananda Sagar University
School of Engineering
Innovation Campus, Kudlu Gate, Hosur Road, Bengaluru

III SEMESTER													
S. N	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	BSC		Transforms and Numerical Techniques	MAT	3	0	0	0	03	60	40	100	**3**
2	IPCC		Data Structures	CSE	3	0	2	0	05	60	40	100	**4**
3	IPCC		Digital Logic Design	ECE	3	0	2	0	05	60	40	100	**4**
4	PCC		Discrete Mathematics and Graph Theory	CSE	3	0	0	0	03	60	40	100	**3**
5	PCC		Foundations of Data Science	DS	3	0	2	0	05	60	40	100	**4**
6	AEC		Liberal Studies	All Dept.	1	0	0	0	01	50	--	50	**1**
7	SEC		Skill Enhancement Course – I	DS	1	0	2	0	03	100	--	100	**2**
			Total		**17**	**0**	**08**	**0**	**25**	**450**	**200**	**650**	**21**

Liberal Studies	
Course Code	**Course Name**
	Drama
	Dance
	Music
	Photography
	Introduction to Japanese language
	Law for Engineers
	Canvas Painting
	Communication in Sanskrit
	Vedic Mathematics
	Critical Thinking
	Introduction to Film Studies
	Yoga & Meditation
	Cyber Crimes, Policies & Laws

Skill Enhancement Course – I	
1	Programming in C++
2	Data Analytics and Visualization Tools (ALTAIR)

IV SEMESTER														
S.N	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week			Examination						Credits
					Lecture	Tutorial	Practical	Project	Duration in Hours	CIE Marks	SEE Marks	Total Marks		
					L	T	P	J						
1	BSC		Probability & Statistics	MAT	3	0	0	0	03	60	40	100		**3**
2	IPCC		Design and Analysis of Algorithms	CSE	3	0	2	0	05	60	40	100		**4**
3	PCC		Database Management System	CSE	3	0	2	0	05	60	40	100		**4**
4	PCC		Web Technologies	DS	3	0	2	0	05	60	40	100		**4**
5	IPCC		Computer Organization and Architecture	CSE	3	0	0	0	03	60	40	100		**3**
6	AEC		Special Topics	DS	0	0	0	04	04	100	--	100		**2**
7	SEC		Skill Enhancement Course – II	DS	1	0	2	0	03	100	--	100		**2**
			Total		**16**	**00**	**08**	**04**	**28**	**500**	**200**	**700**		**22**

Skill Enhancement Course – II	
1	OOPS with JAVA
2	Python Scripting

V SEMESTER													
S.N	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	IPCC / PCC		Theory of Computation	DS	3	1	0	0	04	60	40	100	**4**
2	IPCC / PCC		Software Engineering & Project Management	DS	3	0	0	0	03	60	40	100	**3**
3	IPCC / PCC		Machine Learning Tools and Techniques	DS	3	0	2	0	05	60	40	100	**4**
4	IPCC / PCC		Data Warehouse and Knowledge Mining	DS	3	0	0	0	03	60	40	100	**3**
5	IPCC / PCC		Operating System	DS	3	0	2	0	05	60	40	100	**4**
6	PEC		Professional Elective Course – I	DS	3	0	0	0	03	60	40	100	**3**
7	SEC		Skill Enhancement Course – III	DS	1	0	2	0	03	100	--	100	**2**
			Total		**19**	**01**	**06**	**0**	**26**	**460**	**240**	**700**	**23**

Skill Enhancement Course – III	
1	MATLAB Programming
2	Shell Scripting (LINUX)
3.	Business Intelligence Tools

VI SEMESTER													
S. N	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	HSMC		Management and Entrepreneurship	DS	2	0	0	0	02	60	40	100	**2**
2	IPCC / PCC		Computer Networks	DS	3	0	0	0	03	60	40	100	**3**
3	IPCC / PCC		Compiler Design and System Software	DS	3	0	2	0	05	60	40	100	**4**
4	PCC		Deep Learning	DS	3	0	2	0	05	60	40	100	**4**
5	OEC		Open Elective – I		3	0	0	0	03	60	40	100	**3**
6	PEC		Professional Elective Course – II	DS	3	0	0	0	03	60	40	100	**3**
7	PEC		Professional Elective Course – III/MooC	DS	3	0	0	0	03	60	40	100	**3**
8	PROJ		Minor Project	DS	0	0	0	4	04	100	--	100	**2**
			Total		**20**	**0**	**04**	**04**	**28**	**520**	**280**	**800**	**24**

	VII SEMESTER												
S. N	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	HSMC		Artificial Intelligence	DS	2	0	0	0	03	60	40	100	3
2	OEC		Open Elective – II		3	0	0	0	03	60	40	100	**3**
3	PEC		Professional Elective Course – IV	DS	3	0	0	0	03	60	40	100	**3**
4	PEC		Professional Elective Course – V	DS	3	0	0	0	03	60	40	100	**3**
5	PROJ		Major Project Phase-I	DS	0	0	0	6	03	100	--	100	**3**
			Total		**11**	**0**	**0**	**6**	**15**	**340**	**160**	**500**	**15**

	VIII SEMESTER												
S. N	Course Type	Course Code	Course Name	Teaching Department	Teaching Hours / Week				Examination				Credits
					Lecture	Tutorial	Practical	Project	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1	PEC		Fundamentals of Economics	DS	2	0	0	0	02	60	40	100	**2**
2	PROJ		Major Project Phase-II	DS	0	0	0	22	03	60	40	100	11
3	INT		Research/ Industry Internship – III	DS	0	0	6	0	03	--	100	100	03
			Total		**2**	**0**	**6**	**22**	**08**	**120**	**180**	**300**	**16**

I -21
II -21
III -21
IV - 22
V -23
VI -24
VII - 15
VIII – 16

NOTE: Total Credits (I-Sem to VIII Sem) = 163 credits.

NOTE: Professional elective courses domain-wise

S. N	Domain-wise Clusters		PROFESSIONAL ELECTIVE COURSES				
			PEC-I	PEC-II	PEC-III	PEC-IV	PEC-V
			5th Semester	6th Semester		7th Semester	
1	Domain-1	Course Code					
		Core Data Science	Emerging Trends in Data Science	Data Analytics with Hadoop	Pattern Recognition	Soft Computing	Business Intelligence
2	Domain-2	Course Code					
		Gaming	Computer Graphics	Computer Vision	Augmented Reality and Virtual Reality	High Performance Computing	Game Theory
3	Domain-3	Course Code					
		Cyber Security	Cloud Computing	Introduction to Cyber Security	Digital Forensics	Block chain and Cryptocurrency	Data Privacy and Cyber Security
4	Domain-4	Course Code					
		Automation	Sensor Analytics	Embedded Systems	Natural Language Processing	Social Network Analysis	Information Retrieval
5	Domain-5	Course Code					
		MooC	MooC	MooC	MooC	MooC	MooC

OPEN ELECTIVE COURSES			
Open Elective -I		**Open Elective-II**	
Course Code	**Course Name**	**Course Code**	**Course Name**
	Statistical Tools and Techniques of Data Science		Business Analytics

TRANSFORMS AND NUMERICAL TECHNIQUES
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – III

Course Code	:	**Credits**	: 03
Hours / Week	: 03 Hours	**Total Hours**	: 39 Hours
L–T–P–J	: 3–0–0–0		

Course Learning Objectives:

This Course will enable students to:

1. **Apply** their knowledge of Laplace transforms and inverse Laplace transforms to proficiently solve linear ordinary differential equations with constant coefficients, facilitating the analysis and modelling of complex systems.
2. **Analyze** periodic functions using Fourier series, assessing the convergence properties and precision of the series expansion, thereby enhancing their ability to understand and manipulate periodic phenomena.
3. **Utilize** complex exponential form, Fourier transforms of basic functions, and Fourier sine and cosine transforms to solve problems involving Fourier integrals, developing proficiency in applying these techniques to various mathematical scenarios.
4. **Employ** numerical methods, including Euler's Method, Runge-Kutta 4th order, Adams-Bashforth, and Adams-Moulton Methods, to solve differential equations and effectively analyze dynamic systems, enabling them to model real-world phenomena and make accurate predictions**.**
5. **Apply** finite difference methods, including the Crank-Nicolson method and appropriate techniques for hyperbolic PDEs, to effectively solve different types of partial differential equations (PDEs) such as elliptic, parabolic, and hyperbolic equations, enhancing their problem-solving skills in the context of differential

equations and their applications.

Teaching-Learning Process (General Instructions)
These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I: Laplace Transform and Inverse Laplace Transform	**09 Hours**

Laplace Transforms of Elementary functions (without proof),
(Text Book-1: Chapter 6: 203 to 207).
Laplace Transforms of $e^{at}f(t), t^n f(t)$ *and* $\frac{f(t)}{t}$, Periodic functions, Unit step function and impulse functions
(Text Book-1: Chapter 6:208-230).
Inverse Laplace Transforms- By the method of Partial Fractions, Logarithmic and Trigonometric functions, Convolution Theorem, Inverse Laplace transform using Convolution Theorem ***(Text Book-1: Chapter 6: 238)***.
Solution to Differential Equations by Laplace Transform.
(Text Book-1: Chapter 238-242).

UNIT – II: Fourier Series	**09 Hours**
Periodic Functions, Trigonometric Series ***(Text Book-1: Chapter 11: 495).*** Fourier series Standard function, Functions of any Period 2L, Even and Odd functions, Half-range Expansions. ***(Text Book-1: Chapter 11: 483-492)*** Practical Harmonic analysis (calculate average power and RMS values of periodic waveforms)	
UNIT – III: Fourier Transform	**06 Hours**
Calculation of Fourier integrals using complex exponential form ***(Text Book-1: Chapter 11: 510).*** Fourier transform of basic functions ***(Text Book-1: Chapter 11: 510-516).*** Fourier sine and cosine transforms. ***(Text Book-1: Chapter 11: 518-522).***	
UNIT – IV: Numerical Methods for Solving Ordinary Differential Equations	**07 Hours**
Euler's Method-Basic principles of Euler's method for solving first-order ODEs ***(Text Book-1: Chapter 1:10-12).*** Runge-Kutta 4th order ***(Text Book-1: Chapter 21:904).*** Multistep Methods-Explanation of multistep methods (Adams-Bashforth, Adams–Moulton Methods) ***(Text Book-1: Chapter 21:911-913).*** Second-Order ODE. Mass–Spring System (Euler Method, Runge–Kutta Methods) ***(Text Book-1: Chapter 21:916-918).***	
UNIT – V: Numerical Methods for Partial Differential Equations	**08 Hours**
Classification of PDEs (elliptic, parabolic, hyperbolic), ***(Text Book-1: Chapter 21:922-923).*** Finite Difference Methods (Laplace and Poisson Equations), Derivation of finite difference approximations ***(Text Book-1: Chapter 21:923-927).*** Crank–Nicolson Method ***(Text Book-1: Chapter 21:938-941).*** Method for Hyperbolic PDEs ***(Text Book-1: Chapter 21:943-945).***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** Laplace transforms and inverse Laplace transforms to solve linear ordinary differential equations with constant coefficients, demonstrating proficiency in system analysis and modelling.	L3
2	**Analyze** periodic functions using Fourier series and evaluate the convergence properties and precision of the series expansion.	L2 & L3
3	**Solve** problems involving Fourier integrals by applying complex exponential form, Fourier transforms of basic functions, and Fourier sine and cosine transforms.	L3
4	**Utilize** numerical methods such as Euler's Method, Runge-Kutta 4th order, Adams-Bashforth, and Adams-Moulton Methods to solve differential equations and analyze dynamic systems	L2 & L3
5	**Apply** finite difference methods, including the Crank-Nicolson method and appropriate techniques for hyperbolic PDEs, to solve various types of partial differential equations (PDEs) such as elliptic, parabolic, and hyperbolic equations.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	2	1					1					
CO2	3	2	2						1					
CO3	3	2	2	1					1					
CO4	3	2	2	1					1					

CO5	3	2	2	1					1					

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Erwin Kreyszig, Advanced Engineering Mathematics, 2015, 10th Edition, Wiley India.

REFERENCE BOOKS:

1. Higher Engineering Mathematics, B.S. Grewal, 2015, 43rd Edition, Khanna Publishers.
2. Higher Engineering Mathematics, John Bird, 2017, 6 th Edition, Elsevier Limited.

E-Resources:

1. https://nptel.ac.in/courses/111106139
2. https://nptel.ac.in/courses/111101164
3. https://nptel.ac.in/courses/111105038

****

DATA STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	39(Th)+26(P) Hours
L–T–P-J	:	3–0–2-0			

Prerequisites:

Proficiency in a C programming language.

Course Objectives:

This Course will enable students to:

1. **Understand** the basic approaches for analysing and designing data structures.
2. **Introduce** dynamic memory allocation and C language concepts required for building data structures
3. **Develop** essential skills to construct data structures to store and retrieve data quickly and **efficiently**.
4. **Utilize** different data structures that support different sets of operations which are suitable for various applications.
5. **Explore & Implement** how to insert, delete, search and modify data in any data structure- Stack, Queues, Lists, Trees.
6. **Develop** applications using the available data structure as part of the course for mini-project.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.

3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**
IINTRODUCTION: Introduction to Data Structure, Classification, C Structure and Union, Array Definition, Representation, Operations (Insertion, Deletion, Search and Traversal), Two/Multidimensional Arrays, sparse matrix, C Pointers TB1: 1.1, 2.2, 2.5 ; TB2: 1.1, 1.2,1.3.1-1.3.4; RB1: 5.1 – 5.12, 6.4	
UNIT – II	**09 Hours**
INTRODUCTION TO ADT: **Stack:** Definition, Array Representation of Stack, Operations on Stacks. **Applications of Stack:** Expression evaluation, Conversion of Infix to Postfix, Infix to Prefix Recursion, Tower of Hanoi **Queue:** Definition, Representation of Queues, Operations of Queues, Circular Queue. **Applications of Queue:** Job Scheduling, A Maze Problem TB1: 3.1,3.2, 3.3,3.4,3.5 ; TB2: 2.1, 2.2, 2.3, 3.2, 3.3	
UNIT – III	**09 Hours**
DYNAMIC DATA STRUCTURES: **Linked List:** Types, Representation of Linked Lists in Memory. Traversing, Searching, Insertion & Deletion from Linked List. Circular List, Doubly Linked List, Operations on Doubly Linked List (Insertion, Deletion, Traversal). **Applications:** Stack & Queue Implementation using Linked Lists. **Case Study**: Josephus problem. TB2: 4.2,4.3,4.5	

UNIT – IV	**08 Hours**
TREES: Basic Terminology, Binary Trees and their representation, Complete Binary Trees, Binary Search Trees, Threaded Binary Trees, Operations on Binary Trees (Insertion, Deletion, Search & Traversal). TB1: 5.1,5.2,5.3,5.5,5.7 **Applications:** Expression Evaluation **Case Study:** Game Tree TB2: 5.5.3,5.5.4,5.6	
UNIT – V	**05 Hours**
Efficient Binary Search Trees: Optimal Binary Search Trees, AVL Trees, Red Black Trees, Splay Trees. **Case Study**: B Trees TB1: 10.1,10.2,10.3,10.4, 11.2	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Demonstrate** the key C programming concepts such as pointers, structures, unions and arrays data structures to perform operations such as insertion, deletion, searching, sorting, and traversing.	L3
2	**Utilize** the fundamental concepts of stacks and queues to solve the standard applications like tower of Hanoi, conversion and evaluation of expressions, job scheduling and maze.	L3
3	**Implement** Singly Linked List, Doubly Linked List, Circular Linked Lists, stacks and queues using linked list**.**	L3

4	**Develop** critical thinking and problem-solving skills by designing and implementing efficient algorithms for Non-linear tree data structure and perform insertion, deletion, search and traversal **operations** on it.	L3
5	**Apply** advanced techniques, such as balancing algorithms for AVL trees, Splay trees and Red-Black trees to maintain the balance and efficiency of binary trees.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3												2	
CO2	3		3									2	2	
CO3	3		3									2	2	
CO4	3	2	3									2	2	
CO5	3	2	3									2	2	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS (TB):

1. Ellis Horowitz, Susan Anderson-Freed, and Sartaj Sahni, "Fundamentals of Data structures in C", 2nd Edition, Orient Longman, 2008.
2. A.M. Tannenbaum, Y Langsam, M J Augentien "Data Structures using C", 1st Edition, Pearson, 2019.

REFERENCE BOOKS:

1. Brian. W. Kernighan, Dennis. M. Ritchie, "The C Programming Language", 2nd Edition, Prentice-Hall, 1988.
2. Gilbert & Forouzan, "Data Structures: A Pseudo-code approach with C", 2nd Edition, Cengage Learning,2014.
3. Jean-Paul Tremblay & Paul G. Sorenson, "An Introduction to Data Structures with Applications", 2nd Edition, McGraw Hill, 2013.
4. R.L. Kruse, B.P. Learly, C.L. Tondo, "Data Structure and Program design in C", 5th Edition, PHI ,2009.

E-Resources:

1. https://nptel.ac.in/courses/106102064
2. https://www.coursera.org/learn/data-structures?specialization=data-structures-algorithms
3. https://www.udemy.com/topic/data-structures/free/
4. https://www.mygreatlearning.com/academy/learn-for-free/courses/data-structures
5. https://cse01-iiith.vlabs.ac.in/
6. https://kremlin.cc/k&r.pdf

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving using group discussion.
2. Role play E.g., Stack, Queue, etc.,
3. Demonstration of solution to a problem through programming.
4. Flip class activity E.g., arrays, pointers, dynamic memory allocation, etc.,

LABORATORY EXPERIMENTS

Total Contact Hours: 26

Following are experiments to be carried out using either C programming language

1. To Implement C programs with concepts of pointers, structures.
2. To implement multidimensional array Matrix Multiplication.

3. To search elements in data structure with different search methods.
4. To implement stack, queue and their variations using arrays.
5. To implement stack, queue and their variations using singly linked lists
6. To implement conversion & evaluation of expression using stacks.
7. To Implement doubly circular Linked Lists and variations and use them to store data and perform operations on it.
8. To Implement Addition/multiplication of 2 polynomial using linked lists
9. To implement binary tree traversal techniques.

OPEN-ENDED EXPERIMENTS

1. A man in an automobile search for another man who is located at some point of a certain road. He starts at a given point and knows in advance the probability that the second man is at any given point of the road. Since the man being sought might be in either direction from the starting point, the searcher will, in general, must turn around many times before finding his target. How does he search to minimize the expected distance travelled? When can this minimum expectation be achieved?
2. The computing resources of a cloud are pooled and allocated according to customer demand. This has led to increased use of energy on the part of the service providers due to the need to maintain the computing infrastructure. What data structure will you use for allocating resources which addresses the issue of energy saving? Why? Design the solution.
3. Mini-Project on applying suitable data structure to a given real-world problem.

DIGITAL LOGIC DESIGN

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	39(Th) + 26(P) Hours
L–T–P-J	:	3–0–2-0			

Course Learning Objectives:

This Course will enable students to:

1. **Translate** the elements of digital logic functions to digital system abstractions using Verilog.
2. **Illustrate** simplification of Boolean expressions using Karnaugh
3. **Mode**l combinational logic circuits for arithmetic operations and logical operations
4. **Analyse** and model sequential elements flip-flops, counter, shift registers.
5. **Outline** the concept of Mealy Model, Moore Model and apply FSM to solve a given design problem.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.

7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**

INTRODUCTION:
Number System- Binary, Hexa, Decimal, Octal and its conversion. Canonical Notation - SOP & POS forms, Minimization of SOP and POS forms.
(Text Book-1: Chapter 1: 1.2 to 1.4, Chapter 2: 2.6)
ARITHMETIC CIRCUITS AND VERILOG MODELLING
Adders: Half adder, full adder, Ripple carry adder, parallel adder /subtractor, fast adders-CLA, comparator- 2 bit. Simplification using K-Maps
(Text Book-2: Chapter 5: 5.2, 5.3.3, 5.4,5.5.2, 5.5.3)
Introduction to Verilog, Syntax of Verilog coding, Modelling styles in Verilog, Verilog Operators, Test bench for simulation
(Text Book-3: Chapter 1: 1.1, 1.2.2, 1.3.1, 1.3.2, 1.3.3, 1.4.2, 1.5.1.2, 1.5.2.2, 1.5.3.2, 1.5.4.2, 1.6.2)

UNIT – II	**07 Hours**

Combinational Circuit Building Multiplexers 4:1, 8:1, decoders 3:8, 2:4, demultiplexers 1:4, encoders 8:3, 4:2, code converters- B to G and G to B- Simplification using K-Maps
Verilog for combinational circuits, if else, case-casex, casez, for loop, generate.
(Text Book-2: Chapter 6: 6.1, 6.2, 6.3, 6.4, 6.6)

UNIT – III	**08 Hours**

Sequential Circuits-1
Basic Latch, Gated latches, Flip Flops SR, D, JK, T, master-slave flip-flops JK, Characteristic equations, 0's and 1's Catching Problem, Race round condition, Switch debounce, shift registers- SISO, SIPO, PISO, PIPO, Setup time, Hold time, Propagation Delay
(Text Book-2: Chapter 7: 7.1, 7.2,7.3, 7.4,7.5,7.6, 7.8)

UNIT – IV	**08 Hours**

Sequential Circuits-2
Binary counters – asynchronous and synchronous, mod-n counter, ripple counter- 4 bit.
Verilog blocking and non-blocking,
Mealy Model, Moore Model, State machine notation, Construction of Finite State Machine.
(Text Book-2: Chapter 7: 7.9, 7.11, 7.12.3, 7.12.4, 8.1, 8.2, 8.3, 8.4)

UNIT – V	**08 Hours**

Introduction to Electronic Design Automation:
FPGA Design Flow, ASIC Design flow, architectural design, logic design, simulation, verification and testing, 3000 Series FPGA architecture.
Applications:
Design 4 Bit ALU, 7 Segment display, Vending Machine, 3 Pipeline.
(Text Book-4: Chapter 1)

Laboratory Experiments

Experiments are conducted using Verilog tool /Kits

1.	Introduction to Xilinx tool, FPGA flow
2.	Adder – HA, FA using data flow and behaviour modelling styles
3.	Adder – HA, FA using structural modelling style
4.	Combinational designs – I (blocking and non-blocking/looping examples) **a.** Multiplexer: 4:1, 8:1 MUX. **b.** De Multiplexer: 1:4, 1:8 DEMUX.
5.	Combinational designs – II (different types of case statements) **c.** Encoder with and without Priority: 8:3 and 4:2. **d.** Decoder: 3:8 and 2:4.
6.	Design of 4-bit ALU
7.	Flip Flop: D FF, T FF, JK FF
8.	Design of Mod – n Up/Down Counter with Synchronous reset
9.	Design of Mod – n Up/Down Counter with Asynchronous reset.
10.	Design of Universal shift Register using FSM

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Interpret** Boolean Expressions of digital design in simplified form	L2
2	**Build** the various elements of digital logic system with Verilog	L3
3	**Construct** Combinational and Sequential logic circuits	L3
4	**Analyse** the hardware model of a digital system at different levels of abstraction in Verilog	L4
5	**Evaluate** the functionality of digital design by implementing on FPGA kits	L5
6	**Design digital** systems using FSM	L3

Table: Mapping Levels of COs to POs / PSOs

Cos	Program Outcomes (POs)												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	-	1	-	-	-	-	-	-	-	1	-	1	-	-
CO2	3	2	1	2	3	-	-	-	1	-	1	1	2	1	-
CO3	3	2	3	1	2	-	-	1	1	-	1	1	2	1	-
CO4	3	3	2	3	3	1	-	1	-	1	2	1	2	2	1
CO5	3	3	2	3	3	1	-	-	-	1	-	-	2	2	1
CO6	3	3	3	3	3	2	-	1	2	2	2	2	2	1	2

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. M. Morris Mano Michael D. Ciletti , "Digital Design with an Introduction to the Verilog HDL", 6th Edition, Pearson Education, 2014.

2. Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with Verilog design", McGraw Hill, 2014.

3. Nazein M. Botros, "HDL programming (VHDL and Verilog)", Dreamtech Press, 2006.

4. Douglas J Smith, "HDL Chip Design", Doone publications 1996.

REFERENCE BOOKS:

1. John M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2014.

2. Donald D. Givone, "Digital Principles and Design", McGraw Hill, 2015.

3. Samir Palnitkar, "Verilog HDL: A Guide to Digital Design and Synthesis", Pearson Education, 2016.

E-Resources:

1. https://archive.nptel.ac.in/courses/106/105/106105165/
2. https://nptel.ac.in/courses/117105080

Activity Based Learning (Suggested Activities in Class)

1. Design problem solving and Programming using group discussion. E.g., Traffic light controller, Digital Clock, Elevator.
2. Demonstration of solution to a problem through simulation.

DISCRETE MATHEMATICS AND GRAPH THEORY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	39Hours
L–T–P–J	:	3–0–0–0			

Course Learning Objectives:

This Course will enable students to:

1. **Learn** the set theoretic concept and its application in theory of computation.
2. **Determine** the concepts of mathematical induction, recursive relations and their application.
3. **Illustrate** the association of functions, relations, partial ordered set and lattices with problems related to theoretical computer science and network models.
4. **Discuss** the basics of graph theory and its application in computer networks. Learn the concepts of counting techniques and its application.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teacher can use to accelerate the attainmentof the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture method, but different type of teaching methods may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** Adopt the Active learning that includes brainstorming, discussing, group work, focused listening, formulating questions, notetaking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinkingskills such as the ability to evaluate, generalize, and analyse information rather than simplyrecall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come upwith their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**

SET THEORY: Sets and subsets, Operations on Sets: Basic set operations, algebraic properties of sets, The Addition Principle **RELATIONS AND ITS PROPERTIES:** Relations and their properties, N-Ary Relations and their applications, Representing relations. ***Textbook – 2: 1.1, 1.2 ; Textbook – 1: 7.1., 7.2, 7.3***	
UNIT – II	**06 Hours**
RELATIONS AND ORDER RELATIONS: Closure of relations, Equivalence Relations, Partial Orderings, Functions, The Growth of Functions. **Self-Study:** Transitive Closure and Warshall's Algorithm. ***Textbook – 1: 7.4., 7.5, 7.6, 3.2***	
UNIT – III	**08 Hours**
MATHEMATICAL INDUCTION AND RECURSION: Mathematical Induction, Recurrence Relations:Rabbits and the Fibonacci Numbers, The Tower of Hanoi, Code word Enumeration, Solving LinearRecurrence Relations **Self-Study:** Basic Connectives and Truth Tables ***Textbook-1: 4.1;6.1, 6.2;*1.1**	
UNIT – IV	**09 Hours**
GRAPH THEORY: Graphs and Graph Models. Graph Terminology and Special Types of Graphs: Basic Terminology, Some Special Simple Graphs, Bipartite Graphs, Complete Bipartite Graphs. Representing Graphs and graph isomorphism: Adjacency lists, Adjacency Matrices, Incidence Matrices, Connectivity: Paths, Connectedness in Undirected and Directed Graphs, Vertex and Edge connectivity and their applications. ***Textbook-1: 8.1, 8.2, 8.3, 8.4***	
UNIT – V	**08 Hours**
GRAPHS AND ITS APPLICATIONS: Euler and Hamilton Paths and their applications, Planar Graphs and their Applications, Graph Coloring and its applications. ***Textbook-1: 8.5, 8.7, 8.8***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Identify** the membership of the Set and Relations and perform basic Algebraic operations	L3
2	**Illustrate** the concept of Mathematical Induction and create linear recurrence relations for the given problem	L4

3	**Construct** different types of graphs based on the properties and the real time applications of graph theoretical concepts	L3
4	**Analyze** the methods for optimizing the solution for graph coloring problem, Eulerian and Hamiltonian circuits/planes	L4

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

XT BOOKS:

1. Kenneth H. Rosen, "Discrete Mathematics and its applications", Tata McGraw Hill, 2003.
2. Bernard Kolman, Robert C. Busby, Sharon Ross, "Discrete Mathematical Structures", 3rdEdition, PHI 2001.

REFERENCE BOOKS:

1. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", IV Edition, PearsonEducation, Asia, 2002.
2. J. P. Tremblay, R. Manohar, "Discrete Mathematical Structures with applications tocomputer Science", Tata McGraw Hill, 1987.
3. J K Sharma, "Discrete Mathematics", 3rd edition, 2013, Macmillan India Ltd.

E-Resources:

1. Discrete Mathematics with Algorithms by M. O. Albertson, J. P. Hutchinson - J. 1988,Wiley.
2. Discrete Mathematics for Computer Science, Gary Haggard, John Schlipf, Sue Whitesides,Thomson Brooks/Cole, 2006.

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	1	2					1	1	1		2	2	1
CO2	3	3	2					1	1	1		2	2	1
CO3	3	3	3					1	1	1		1	2	1
CO4	3	3	3					1	1	1		2	2	1
Avg	3	2.5	2.5					1	1	1		1.75	2	1

3. http://ocw.mit.edu/courses/mathematics/
4. http://www.nptelvideos.in/2012/11/discrete-mathematical-structures.html
5. http://cglab.ca/~discmath/notes.html
6. https://www.cs.odu.edu/~toida/nerzic/content/web_course.html

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion.
2. Demonstration of solution to a problem using graph theory.

****
******

FOUNDATIONS OF DATA SCIENCE
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	65 Hours
L–T–P	:	3–0–2-0			

Prerequisites:

Proficiency in a Python and R programming language.

Course Learning Objectives:

This Course will enable students to:

1. **Familiarize** with Python libraries, specifically for data science tasks, using Pycharm as the development environment
2. **Gain** knowledge of parametric tests, such as the Z-test, one-sample T-test, paired T-test, independent sample T-test, ANOVA, MANOVA, and their significance levels and power values. Also, **learn** about non-parametric tests, including the chi-square test, Fisher's test, Mann-Whitney U test, Kruskal-Wallis rank test, and Wilcoxon sign rank test.
3. **Study** classification models, including logistic regression, discriminant regression analysis, support vector machines (SVM), naive Bayes, random forests, CHAID analysis, decision trees, k-nearest neighbors, and neural networks. **Understand** their principles, strength of associations, maximum likelihood estimation, and the use of confusion matrices.
4. **Explore** unsupervised learning techniques, such as principal component analysis (PCA), reliability tests, KMO tests, eigenvalue interpretation, and clustering methods like K-means clustering and agglomerative clustering.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.

5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**
INTRODUCTION: Overview of the Data science process. Different types of data Data Pre-processing: Data Cleaning- Missing values, Noisy data. Data cleaning as a process. Data Reduction: principal component analysis. Data Transformation: Strategies overview. Data transformation by normalization. Discretization by binning. (Text Book-1: Chapter1) (Text Book-2: Chapter 3)	
UNIT – II	**09 Hours**
EXPLORATORY DATA ANALYSIS AND HYPOTHESIS TESTING: Exploratory Data Analysis: Central Tendency (Text Book-1: Chapter 3-3.1), Dispersions, Five number Distributions, Cross Tabulations. Data Visualization: Histogram, Box Plot, Correlation Plot, Scatter Plot, Line Chart, Bar Chart, Pie Chart, Bubble Chart, Decision Tree, Cluster Charts.(Text Book-1:Chapter 2.3,2.4,2.5) Hypothesis Testing: Confidence Intervals (Text Book-1:Chapter 8), Constructing a hypothesis, Null Hypothesis; Alternative Hypothesis, Type I and Type II errors, Power Value(Text Book-1: Chapter 9)	
UNIT – III	**08 Hours**
PARAMETRIC AND NON-PARAMETRIC TESTS: Parametric test: Z test, One Sample T-TEST, Paired T-TEST, Independent Sample T-TEST, ANOVA, MANOVA, Level of significance, Power of a test.(Text Book-1:Chapter 9, Chapter 11) Non parametric test: Chi Square Test, Kruskal-Wallis Rank Test, Wilcoxon sign rank.(Text Book-1: Chapter 12)	
UNIT – IV	**07 Hours**
Overview of R- Basic Features of R, R Conventions- R for Basic Math- Arithmetic- Logarithms and Exponentials, E-Notation- Assigning Objects- Vectors- Creating a Vector- Sequences, Repetition, Sorting, and Lengths- Subsetting and Element Extraction- Vector-Oriented Behaviour. Defining a Matrix – Defining a Matrix- Filling Direction- Row and Column Bindings- Matrix DimensionsSubsetting- Row, Column, and Diagonal Extractions- Omitting and Overwriting- Matrix Operations and Algebra	

UNIT – V	**07 Hours**
Advanced R- Lists of Objects-Component Access-Naming-Nesting-Data Frames-Adding Data Columns and Combining Data Frames-Logical Record Subsets-Some Special Values-Infinity-NaN-NA-NULL, Attributes-Object-Class-Is-Dot Object-Check. Plotting with Coordinate Vectors-Graphical Parameters-Automatic Plot Types-Title and Axis Labels, Color-Line and Point Appearances-Plotting Region Limits-Adding Points, Lines, and Text to an Existing Plot-ggplot2 Package-Quick Plot with qplot-Setting Appearance Constants with Geoms READING AND WRITING FILES- R-Ready Data Sets- Contributed Data Sets- Reading in External Data Files- Writing Out Data Files and Plots- Ad Hoc Object Read/Write Operations	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	To Summarize the data using visual & amp; summary analytics and common probability distributions	L3
2	To make inference about a sample & population using hypothesis tests.	L3
3	To fit, interpret, and assess regression models and classification with one or more predictors.	L3
4	To apply statistical analysis using R Programming	L3

5	Understanding and analyzing the Data using R packages	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	3										2	
CO2	3	3	3										2	
CO3	3	3	3										2	
CO4	3	3	3										2	
CO5	2	3	3										2	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, EMC Education Services, Wiley
2. Tilman M.Davies,"THE BOOK OF R - A FIRST PROGRAMMING AND STATISTICS" Library of Congress Cataloging-in-Publication Data,2016

REFERENCE BOOKS:

1. Data Mining in excel: Lecture Notes and cases by Galit Shmueli, Publisher: Wiley
2. Hastie, Tibshirani, Friedman, "The Elements of Statistical Learning" (2nd ed)., Springer, 2008

E-Resources:

1. https://www.simplilearn.com/pgp-data-science-certification-bootcamp-program

Practical Experiments:

Design a R/Python:

1. Program to import iris dataset and display head, tail, summary, interquartile range and structure.
2. Program for data cleaning and find missing values in iris dataset.
3. Program for exploratory data analysis(EDA)
4. Program to plot correlation matrix and covariance plot for iris dataset.
5. Program for dimensionality reduction using principal component analysis for iris dataset.
6. Implement a program to select required features from iris dataset using chi square test
7. Build a python program for iris dataset using different classification algorithms.
8. Implement a program for Simple Linear Regression and Multiple Linear Regression
9. Program for One-way ANOVA and Two-Way ANOVA.

****

PROBABILITY AND STATISTICS [As per Choice Based Credit System (CBCS) scheme] **SEMESTER – IV**					
Course Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	39 Hours
L–T–P–J	:	3–0–0–0			

Course Learning Objectives:

This Course will enable students to:

1. **Apply** statistical principles and probability concepts to solve complex problems in real-world scenarios involving uncertainty and randomness.
2. **Evaluate** and select appropriate probability distributions and statistical techniques to analyze and interpret data accurately in various applications.
3. **Justify** the use of estimation methods and hypothesis testing techniques for drawing meaningful inferences about population parameters.
4. **Analyze** and interpret sample test results for different statistical relationships, such as means, variances, correlation coefficients, regression coefficients, goodness of fit, and independence, to make informed decisions.
5. **Identify** sample tests using appropriate statistical procedures to investigate the significance of observed data and communicate findings effectively.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I : Probability	**09 Hours**
Definitions of Probability, Addition Theorem, Conditional Probability, Multiplication Theorem, Bayes' Theorem of Probability	
UNIT – II: Random Variables and their Properties and Probability Distributions	**09 Hours**
Discrete Random Variable, Continuous Random Variable, Joint Probability Distributions Their Properties, Probability Distributions: Discrete Distributions: Binomial, Poisson Distributions and their Properties; Continuous Distributions: Exponential ,Normal, Distributions and their Properties.	
UNIT – III: Estimation and testing of hypothesis	**06 Hours**
Sample, Populations, Statistic, Parameter, Sampling Distribution, Standard Error, Unbiasedness, Efficiency, Maximum Likelihood Estimator, Notion & Interval Estimation.	

UNIT – IV: Sample Tests-1	**07 Hours**
Large Sample Tests Based on Normal Distribution , Small Sample Tests : Testing Equality of Means, Testing Equality of Variances, Test of Correlation Coefficient	
UNIT – V: Sample Tests-2	**08 Hours**
Test for Regression Coefficient; Coefficient of Association, 2 – Test for Goodness of Fit, Test for Independence.	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** the principles of probability to solve complex problems in various real-world scenarios.	L2 & L3
2	**Solve** and **compare** different probability distributions, including discrete and continuous random variables, in order to make informed decisions and predictions.	L2 & L3
3	**Apply** statistical estimation techniques, such as maximum likelihood estimation and interval estimation, to draw meaningful inferences about population parameters from sample data.	L3
4	**Examine** hypothesis testing methods, including large and small sample tests, to assess the significance of observed data and draw valid conclusions.	L4
5	**Analyze** statistical relationships and perform sample tests to assess the Equality of means in different populations, Correlation coefficients between variables to determine the strength and direction of the relationship. Independence of variables using appropriate statistical tests to assess the absence of any relationship.	L4

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO 1	3	2	2		2				1					
CO 2	3	2	2		2				1					
CO 3	3	2	2						1					
CO 4	3	2	2		2				1					
CO 5	3	2	2		2				1					

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Probability & Statistics for Engineers and Scientists, Walpole, Myers, Myers, Ye. Pearson Education.

REFERENCE BOOKS:

1. Probability, Statistics and Random Processes T. Veerarajan Tata McGraw – Hill
2. Probability & Statistics with Reliability, Queuing and Computer Applications, Kishor S. Trivedi, Prentice Hall of India ,1999

E-Resources:

1. https://nptel.ac.in/courses/106104233
2. https://nptel.ac.in/courses/117103067
3. https://nptel.ac.in/courses/103106120
4. https://www.coursera.org/learn/probability-intro#syllabus
5. https://nptel.ac.in/courses/111104073

Activity Based Learning (Suggested Activities in Class)

1. Tools like Python programming, R programming can be used which helps students to develop a skill to analyze the problem and provide a solution.
2. Regular Chapter wise assignments/ Activity/Case studies can help students to have critical thinking, developing an expert mind set, problem-solving and teamwork.

Following are Assignments/ Activities Can be carried out using either R programming language or Python Programming or excel solver.

1. There are n people gathered in a room. What is the probability that at least 2 of them will have the same birthday? (Use excel solver, R Programming, Python Programming)
   a. Use simulation to estimate this for various n., and Produce Simulation Graph.
   b. Find the smallest value of n for which the probability of a match is greater than 0.5.
   c. Explore how the number of trials in the simulation affects the variability of our estimates.
2. **Case Study 1: Customer Arrivals at a Coffee Shop**
   a. A coffee shop wants to analyze the number of customer arrivals during its morning rush hour (7:00 AM to 9:00 AM). The shop has been recording the number of customer arrivals every 15 minutes for the past month.
   b. Data: The data consists of the number of customer arrivals recorded at the coffee shop during each 15-minute interval for the past month.
   c. Here is a sample of the data:

Time Interval	ıstomer Arrivals
00 AM - 7:15 AM	**6**
15 AM - 7:30 AM	**4**
30 AM - 7:45 AM	**9**
45 AM - 8:00 AM	**7**
00 AM - 8:15 AM	**5**
15 AM - 8:30 AM	**8**
30 AM - 8:45 AM	**10**
45 AM - 9:00 AM	**6**

   analyze the customer arrivals and determine the probability distribution that best fits the data. Specifically, explore both discrete and continuous probability distributions, including the binomial, Poisson, exponential, and normal distributions.
3. **Case Study 2: Comparing the Performance of Two Groups**
   a. Suppose you are a data analyst working for a company that manufactures a new energy drink. The marketing team conducted a promotional campaign in two

different cities (City A and City B) to determine the effectiveness of the campaign in increasing sales. The sales data for a random sample of customers in each city was collected over a week. Your task is to compare the average sales between the two cities and test whether there is a significant difference in the variance of sales.

b. **Data**: Let's assume the following sample data for the number of energy drinks sold in each city:

City A: [30, 28, 32, 29, 31, 33, 34, 28, 30, 32]
City B: [25, 24, 26, 23, 22, 27, 29, 30, 26, 24]

perform a two-sample t-test to test the equality of means and a test for equality of variances using Python's SciPy library.

4. **case study 3:** testing independence between two categorical variables.
   a. Data: Sample of 100 employees, and each employee is classified as either Male or Female. They were asked to rate their job satisfaction on a scale of 1 to 5, where 1 represents low satisfaction and 5 represents high satisfaction. The data is as follows:

Employee	Gender	Job Satisfaction
1	Male	4
2	Female	3
3	Male	2
4	Female	5
...	...	...
100	Female	4

   b. Test for independence between gender and job satisfaction, use the chi-squared test in R.

****

DESIGN AND ANALYSIS OF ALGORITHMS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	39(Th)+26(P) Hours
L–T–P–J	:	3–0–2–0			

Course Learning Objectives:

This Course will enable students to:

1. **Analyze** the non-recursive and recursive algorithms and to represent efficiency of these algorithms in terms of the standard Asymptotic notations.
2. **Acquire** the knowledge of Brute Force and Divide and Conquer techniques to design the algorithms and apply these methods in designing algorithms to solve a given problem.
3. **Master** the Decrease and Conquer, Transform and Conquer algorithm design techniques, and Time versus Space Trade-offs.
4. **Learn** Greedy methods and dynamic programming methods and apply these methods in designing algorithms to solve a given problem.
5. **Understand** the importance of Backtracking and Branch and Bound algorithm design techniques to solve a given problem.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the

class.

6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**08 Hours**
INTRODUCTION: What is an Algorithm? Fundamentals of Algorithmic Problem Solving. ***(Text Book-1: Chapter 1: 1.1 to 1.2)*** **FUNDAMENTALS OF THE ALGORITHMS EFFICIENCY:** Analysis Framework, Asymptotic Notations and Standard notations and common functions ***(Text Book-2: Chapter 3: 3.1, 3.2)***, Mathematical Analysis of Non-recursive and Recursive Algorithms, ***(Text Book-1: Chapter 2: 2.1, 2.3, 2.4,)***	
UNIT – II	**08 Hours**
BRUTE FORCE: Background, Selection Sort, Brute-Force String Matching. TSP ***(Text Book-1: Chapter 3: 3.1, 3.2)*** **DIVIDE AND CONQUER:** General method, Recurrences: The substitution method, The recursion-tree method, The master method. ***(Text Book-2: Chapter 4: 4.4, 4.5***), Merge sort, Quick sort, Binary Search, Multiplication of large integers, **Case study:** Strassen's Matrix Multiplication. ***(Text Book-1: Chapter 4: 4.1 to 4.3, 4.5)***	
UNIT – III	**06 Hours**
DECREASE & CONQUER: General method, Insertion Sort, Graph algorithms: Depth First Search, Breadth First Search, Topological Sorting **TRANSFORM AND CONQUER:**	

Case study: Heaps and Heap sort.

TIME AND SPACE TRADEOFFS:
Input Enhancement in String Matching: Horspool's algorithm, Hashing: Open and Closed hashing.
(Text Book-1: Chapter 5: 5.1 to 5.3, Chapter 6: 6.3 to 6.4, Chapter 7: 7.2 to 7.3)

UNIT – IV	**9 Hours**

GREEDY TECHNIQUE:
General method of Greedy technique, Single-Source Shortest Paths: General method, The Bellman-Ford algorithm, Single-Source Shortest Paths in DAGs, Dijkstra's Algorithm
(Text Book-2: Chapter 24: 24.1 to 24.3).
Minimum Spanning Trees: Prim's Algorithm, Optimal Tree problem: Huffman Trees;
Case study: Kruskal's Algorithm. Fractional Problem
(Text Book-1: Chapter 9: 9.1, 9.2, 9.4).
DYNAMIC PROGRAMMING:
General method, The Floyd-Warshall Algorithm, Johnson's algorithm for sparse graphs ***(Text Book-2: Chapter 25: 25.1 to 25.3)***,
The Knapsack problem ***(Text Book-1: Chapter 8: 8.4).***

UNIT – V	**08 Hours**

LIMITATIONS OF ALGORITHMIC POWER
P, NP and NP-complete problems ***(Text Book-1: Chapter 11: 11.3***)
BACKTRACKING:
General method, N-Queens problem, Subset-sum problem.
(Text Book-1: Chapter 12: 12.1)
BRANCH AND BOUND:
General method, Travelling Salesman problem, Approximation algorithms for TSP.
Case study: Knapsack Problem.
(Text Book-1: Chapter 12: 12.2, 12.3)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Exemplify** the algorithm design techniques and standard Asymptotic notations. Analyze non-recursive and recursive algorithms to obtain worst-case running times of algorithms using asymptotic analysis	L3
2	**Interpret** the brute-force, divide-and-conquer paradigms and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.	L3
3	**Demonstrate** the Decrease and Conquer, Transform and Conquer algorithm design techniques and analyze the performance of these algorithms.	L3
4	**Identify and interpret** the greedy technique, dynamic-programming paradigm as to when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms and analyze them	L3
5	**Illustrate** the Backtracking, Branch and Bound algorithm design paradigms and explain when an algorithmic design situation calls for it. Recite algorithms that employ these paradigms. **Summarize** the limitations of algorithmic power.	L3

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO 1	3	3										2		3
CO 2	3	3	2									2		3
CO 3	3	3										1		3
CO 4	3	3	2									2		3
CO 5	3	3										1		3

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Anany Levitin, "Introduction to the Design & Analysis of Algorithms", 2nd Edition, Pearson Education, 2011.
2. Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI, 2014.

REFERENCE BOOKS:

1. Horowitz E., Sahni S., Rajasekaran S, "Computer Algorithms", Galgotia Publications, 2001.
2. R.C.T. Lee, S.S. Tseng, R.C. Chang & Y.T.Tsai, "Introduction to the Design and Analysis of Algorithms A Strategic Approach", Tata McGraw Hill, 2005.

E-Resources:

1. https://nptel.ac.in/courses/106/101/106101060/
2. http://cse01-iiith.vlabs.ac.in/
3. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms
4. https://www.coursera.org/specializations/algorithms

Activity Based Learning (Suggested Activities in Class)

1. Real world problem solving and puzzles using group discussion. E.g., Fake coin identification, Cabbage puzzle, Konigsberg bridge puzzle etc.,
2. Demonstration of solutions to a problem through programming.

LABORATORY EXPERIMENTS

Total Contact Hours: 26

Following are experiments to be carried out using either C programming language or Object-oriented programming language:

1. Apply divide and conquer method and Design a C program to implementation of Binary Search algorithm.
2. Sort a given set of n integer elements using the Merge Sort method and compute its time complexity. Demonstrate this algorithm using the Divide-and-Conquer method.
3. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Demonstrate this algorithm using the Divide-and-Conquer method.
4. Incorporate the array data structure and demonstrate whether a given unweighted graph is connected or not using the DFS method.
5. Implement the graph traversal technique using the BFS method to print all the nodes reachable from a given starting node in an unweighted graph.
6. Compute the Transitive Closure for a given directed graph using Warshall's algorithm.
7. For a given weighted graph, construct an All-Pairs Shortest Paths problem using Floyd's algorithm and implement this algorithm to find the shortest distance and their shortest paths for every pair of vertices.
8. Implement 0/1 Knapsack problem using Dynamic Programming Memory Functions technique
9. Find Minimum Cost Spanning Tree for a given weighted graph using Prim's and Kruskal's algorithm.
10. From a given vertex in a weighted connected graph, determine the Single Source Shortest Paths using Dijkstra's algorithm.
11. Mini project proposal should be submitted and Implementation should be done based on the problem stated in the proposal

Open ended experiments

1. Implement Fractional Knapsack problem using Greedy Method.
2. Implement N-Queens problem using Backtracking technique.
3. implementation of Traveling Salesman problem using Dynamic programming

DATABASE MANAGEMENT SYSTEM

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	:		**Credits**	:	04
Hours / Week	:	03 Hours	**Total Hours**	:	39(Th)+26(P) Hours
L–T–P–J	:	3–0–2–0			

Course Learning Objectives:

This course will enable students to:

1. **Acquire** the concept of databases, Entity-Relationship Model and relational model for creating and designing databases for the real-world scenario.
2. **Develop** queries to extract data from the databases using a structured query language.
3. **Differentiate** SQL and NoSQL.
4. **Demonstrate** the operations on MongoDB, Database connectivity with front end and **Optimize** the Database design using Normalization Concepts.
5. **Understand** the importance of Transaction Management, Concurrency control mechanism and recovery techniques.

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods that teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only the traditional lecture method but a different *type of teaching method* that may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note-taking, annotating, and roleplaying.
3. Show ***Video/animation*** films to explain the functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher-order Thinking questions in the class.

6. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the student's understanding.	
UNIT – I	**10 Hours**
INTRODUCTION TO DATABASE SYSTEMS : Introduction, Characteristics of the Database Approach, Advantages of using DBMS Approach, Data Models, Schemas, Instances and Data Independence, Three Schema Architecture, various components of a DBMS. ***(Text Book-1: Chapter 1: 1.1 to 1.4, 1.6, Chapter 2: 2.1,2.2, 2.4)*** **ENTITY-RELATIONSHIP MODEL:** Entity Types , Entity Sets , Attributes and Keys, Relationship types, Relationship Sets , Roles and Structural Constraints; Weak Entity Types; ER Diagrams ***(Text Book-1: Chapter 7: 7.3, 7.4, 7.5, 7.7)***.	
UNIT – II	**07 Hours**
RELATIONAL MODEL : Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update operations and Dealing with Constraint Violations. ***(Text Book-1: Chapter 3: 3.1 to 3.3).*** **SQL –THE RELATIONAL DATABASE STANDARD:** SQL Data Definition and Data types, Specifying constraints in SQL, Basic Queries in SQL-Data Definition Language in SQL, Data Manipulation Language in SQL; ***(Text Book-1: Chapter 4: 4.1 to 4.4).***	
UNIT – III	**08 Hours**
SQL –THE RELATIONAL DATABASE STANDARD: Additional Features of SQL; Views (Virtual Tables) in SQL; Database Programming Issues and Techniques ; ***(Text Book-1: Chapter 4: 4.5; Chapter 5: 5.1 to 5.4).*** **SQL AND NOSQL DATA MANAGEMENT:** Triggers, Database connectivity using Python, SQL vs NoSQL, Introduction to MongoDB, ***(Text Book-1: Chapter 5: 5.2,5.3***)***(Text Book-2 Chapter 1: 1.1 to 1.5***)	
UNIT – IV	**07 Hours**
NOSQL DATA MANAGEMENT: Data Types, Data Modelling, CRUD Operations. ***(Text Book-2 Chapter 1: 1.1 to 1.5***)	

DATABASE DESIGN: Design Guidelines, Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form; ***(Text Book-1: Chapter 14: 14.1 to 14.5)***
UNIT – V **07 Hours**
TRANSACTION MANAGEMENT The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Concurrency Control Mechanisms; Error recovery methods. ***(Text Book-1: Chapter 20: 20.1 to 20.5, Chapter 21: 21.1 to 21.3, Chapter 22: 22.1 to 22.4***)

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Use** the basic concepts of database management system in the design and creating database blueprint using E-R model and relational model.	L3
2	**Formulate** SQL and NoSQL queries for building structure and unstructured databases	L3
3	**Demonstrate** database connectivity using vendor specific drivers	L3
4	**Apply** normalization techniques to design relational database management system	L3
5	**Adapt** Transaction Management, concurrency control and recovery management techniques in database management systems.	L3

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	3	3	2	-	-	-	-	-	2	2	-	2	3	-
CO2	3	2	1	-	3	-	-	-	2	2	-	2	3	-
CO3	2	2	2	-	3	-	-	-	2	2	-	2	3	-
CO4	3	1	2	-	1	-	-	-	2	2	-	2	3	-
CO5	2	1	-	-	-	-	-	-	2	2	-	2	3	-

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Elmasri and Navathe, ”Fundamentals of Database Systems“, Seventh Edition, Pearson Education, 2021, 2015.
2. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", First Edition, Pearson Education, Inc. 2012.

REFERENCE BOOKS:

1. Raghu Ramakrishnan and Johannes Gehrke, “Database Management Systems”, Third Edition, McGraw-Hill, 2003.
2. Silberschatz, Korth and Sudharshan: “Database System Concepts”, Seventh Edition, Mc-GrawHill, 2019.
3. C.J. Date, A. Kannan, S. Swamynatham: “An Introduction to Database Systems”, Eight Edition, Pearson Education, 2012.

E-Resources:

1. http://nptel.ac.in/courses/106106093/

2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-830-database-systems-fall-2010/lecture-notes/
3. http://agce.sets.edu.in/cse/ebook/DBMS%20BY%20RAGHU%20RAMAKRISHNAN.pdf
4. http://iips.icci.edu.iq/images/exam/databases-ramaz.pdf
5. https://db-class.org/
6. https://www.w3schools.com/mongodb/

Activity Based Learning (Suggested Activities in Class)

1. Database designing and data extraction using group discussion.
2. Collaborative Activity is minor project development with a team of 4 students.

LABORATORY EXPERIMENTS

Total Contact Hours: 26

Following are experiments to be carried out using either oracle or mysql, Mongo Db .

1. Design any database with at least 3 entities and establish proper relationships between them. Draw suitable ER/EER diagrams for the system. Apply DCL and DDL commands.
2. Design and implement a database and apply at least 10 Different DML Queries for the following task.
    a. For a given input string display only those records which match the given pattern or a phrase in the search string. Make use of wild characters and like operators for the same. Make use of Boolean and arithmetic operators wherever necessary
3. Write SQL statements to join tables and retrieve the combined information from tables.
4. Execute the Aggregate functions count, sum, avg, min, max on a suitable database. Make use of built in functions according to the need of the database chosen .
5. Retrieve the data from the database based on time and date functions like now(), date(), day(), time() etc., Use of group by and having clauses.
6. Write and execute database triggers. Consider row level and statement level triggers.
7. Write and execute a program to perform operations on MongoDb Database.
8. Write and execute a program to perform CRUD operations.

Open Ended Experiments

1. Consider the Table “employees”, write a SQL query to remove all the duplicate emails of employees keeping the unique email with the lowest employee id,return employee id and unique emails .

    table: employees

    employee_id |employee_name |email_id |

101	Liam Alton	li.al@abc.com
102	Josh Day	jo.da@abc.com
103	Sean Mann	se.ma@abc.com
104	Evan Blake	ev.bl@abc.com
105	Toby Scott	jo.da@abc.com

2. A salesperson is a person whose job is to sell products or services. Consider the table "Sales" [given below]. Write a SQL query to find the top 10 salespeople that have made the highest sale. Return their names and total sale amount.

Table: sales

TRANSACTION_ID	SALESMAN_ID	SALE_AMOUNT
501	18	5200.00
502	50	5566.00
503	38	8400.00
599	24	16745.00
600	12	14900.00

Table: salesman

SALESMAN_ID	SALESMAN_NAME
11	Jonathan Goodwin
12	Adam Hughes
13	Mark Davenport
59	Cleveland Hart
60	Marion Gregory

Dayananda Sagar University
School of Engineering
Innovation Campus, Kudlu Gate, Hosur Road, Bengaluru

WEB TECHNOLOGIES
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	:			**Credits**	:	03
Hours / Week	:	03 Hours		**Total Hours**	:	39 + 26 Hours
L–T–P	:	3–0–2-0				

Prerequisites: ***NIL***

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the fundamentals of front end web technologies using HTML 5 and CSS3
2. **Apply** Cascading Style Sheets and XHTML to the idea of a web application.
3. **Understand** the principles of client-side programming and understand how to use JavaScript to implement them in order to create dynamic web sites.**Usage** of wide variety of testing techniques in an effective and efficient manner
4. **Implement** the principles of server side programming using Node.js, Mongo dB
5. **Apply** the Node.js framework -Express.JS to create web applications faster and smarter

Teaching-Learning Process (General Instructions)

These are sample new pedagogical methods, where teachers can use to accelerate the attainment of the various course outcomes.

1. ***Lecture method*** means it includes not only traditional lecture methods, but different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *Adopt the* ***Active learning*** that includes brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Show ***Video/Animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, ask at least three Higher order Thinking questions in the class.
6. Adopt ***Problem Based Learning***, which fosters students' Analytical skills, develop thinking skills such as the ability to evaluate, generalize, and analyse information rather than simply recall it.
7. Show the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.
8. Discuss how every ***concept can be applied to the real world*** - and when that's possible, it helps improve the students' understanding.

UNIT – I	**04 Hours**

Mark-up Language (HTML5):

Introduction to HTML and HTML5 - Formatting and Fonts -Commenting Code – Anchors – Backgrounds –(Text Book 1, Chapter 1
Images – Hyperlinks – Lists – Tables – HTML Forms, Audio, Video Tag.

UNIT – II	**04 Hours**

CSS3:

Levels of style sheets; Style specification formats; Selector forms; Property value forms; Font properties; List properties; Color; Alignment of text; Background images, Conflict Resolution, CSS Box Model.CSS3 features: Box Shadow, Opacity, Rounded corners, Attribute selector.

UNIT – III	**06 Hours**

JavaScript:
Overview of JavaScript; Object orientation and JavaScript; General syntactic, characteristics; Primitives, operations, and expressions; Screen output and keyboard input. Control statements; Arrays; Functions, Constructors; A brief introduction on pattern matching using regular expressions, DOM Events

UNIT – IV	**06 Hours**

Node JS:
Introduction to NodeJS, Set up Dev Environment, Node JS Modules, Node Package Manager, File System, Events, Database connectivity using Mongo DB.
AngularJS: Power Features of AngularJS, MVC Architecture: Conceptual Overview, Setting up the Environment, The Anatomy of an AngularJS app, First Application. Number and String Expressions, Object Binding and Expressions, Working with Arrays, Understanding Data binding, Modular Programming, Controllers, Attaching Properties and Functions to Scope, Adding Logic to the Controller, Adding Instance Functions and Properties to Controllers.

UNIT – V	**06 Hours**

Express.JS
Introducing Express: Basics of Express, Express JS Middleware: Serving Static Pages, Listing Directory Contents, Accepting JSON Requests and HTML Form Inputs, Handling Cookies.
React JS Basics & Features, Setup and Hello World Application, Components and Props, Function and Class Components, Rendering Components, Comment Box in React, Handling Events.

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	Develop proficiency in creating structured web pages using HTML5 elements, formatting, fonts, images, hyperlinks, lists, tables, and forms.	L4
2	Gain a thorough understanding of CSS3 for styling web pages, including selectors, properties, fonts, colors, and the CSS box model.	L3
3	Acquire skills in JavaScript for client-side scripting, including control statements, arrays, functions, regular expressions, and DOM events.	L2
4	Understand the fundamentals of Node.js, Angular.js including module usage, file system operations, event handling, and database connectivity with MongoDB.	L2
5	Gain proficiency in building web applications with React, Express.js, including serving static pages, handling JSON requests, listing directory contents, and managing cookies.	L3

Table: Mapping Levels of COs to POs / PSOs

COs	Program Outcomes (POs)												PSOs	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO 1	3		2										2	
CO 2	3		3										2	

CO 3	3	3	1									2	
CO 4	3	3	1									2	
CO 5	3	3	1									2	

3: Substantial (High) **2: Moderate (Medium)** **1: Poor (Low)**

TEXT BOOKS:

1. Robert W. Sebesta , "Programming the World Wide Web", 7th Edition, Pearson Education, 2008.
2. Basarat Ali Syed," Beginning Node.js ",Apress ,2014

E-BOOKS / ONLINE RESOURCES:

1. https://www.geeksforgeeks.org/html/
2. https://www.geeksforgeeks.org/css/
3. https://www.geeksforgeeks.org/javascript/

Activity Based Learning (Suggested Activities in Class)

1. Frontend Development
2. Database Management
3. Deployment and DevOps

List of Lab Programs **26 Hrs**

1. Programs on basics of HTML
2. Programs on CSS with HTML
3. Programs on NodeJS and AngularJS
4. Programs on ReactJS
5. Mini Project on Frontend Design

COMPUTER ORGANIZATION AND ARCHITECTURE
[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Course Code	:		**Credits**	:	03
Hours / Week	:	03 Hours	**Total Hours**	:	39 Hours
L–T–P-J	:	3–0–0-0			

Course Learning Objectives:

This Course will enable students to:

1. **Understand** the Architecture and programming of ARM microprocessors.
2. **Develop** program using Arm instruction set and appreciate the advanced features provided in the ARM
3. **Understand** the exception handling techniques.
4. **Study in** detail the concept of instruction level parallelism and concepts of pipelining.
5. **Understand** various cache memory mapping techniques and memory Organization.

Teaching-Learning Process

1. ***Lecture method*** along with traditional lecture method, different *types of teaching methods* may be adopted to develop the course outcomes.
2. ***Interactive Teaching:*** *incorporating* brainstorming, discussing, group work, focused listening, formulating questions, note taking, annotating, and roleplaying.
3. Showing ***Video/animation*** films to explain functioning of various concepts.
4. Encourage ***Collaborative*** (Group Learning) Learning in the class.
5. To make ***Critical thinking***, asking Higher order Thinking questions in the class in the form of Quiz and writing programs with complex solutions.
6. Showing the ***different ways to solve*** the same problem and encourage the students to come up with their own creative ways to solve them.

UNIT – I	**05 Hours**
An Overview of Computing Systems: History of Computers, The Computing Device, **The ARM7TDMI Programmers' Model:** Introduction, Data types, Processor Modes, Registers, Program Status Registers, The vector Table. **Assembler Rules and Directives:** Structure of Assembly Language Modules, Registers, Directives and Macros. **Loads, Stores and Addressing:** LODS and STORES instructions, Operand Addressing , ENDIANNES ***Text Book-1: 1.1 to 1.3; 2.1 to 2.6 ; 4; 5.3, 5.4, 5.5***	
UNIT – II	**05 Hours**
Constants and Literal Pools: The ARM Rotation Scheme, Loading Constants and address into Registers **Logic and Arithmetic:** Flags and their Use, Compare instructions, Data Processing Instructions **Loops and Branches:** Branching, Looping, Conditional Execution, Straight-Line Coding **Subroutines and Stacks:** Stack, Subroutines, Passing parameters to subroutines, The ARM APCS. ***(Text Book-1: 6.1 to 6.4; 7.1 to 7.4; 8.2 to 8.6; 10.1 to 10.5***	
UNIT – III	**05 Hours**
Mixing C and Assembly Language: Inline Assembler Embedded Assembler, Calling Between C and Assembly. **Exception Handling:** Interrupts, Error Conditions, Processor Exception Sequence, The Vector Table, Exception Handlers, Exception Priorities, Procedures for Handling Exceptions. ***(Text Book-1: 11.1 to 11.8; 14.1 to 14.4***	

UNIT – IV	12 Hours
Pipelining: Basic and Intermediate Concepts Introduction, The Major Hurdle of Pipelining, How Pipelining Implemented, What makes Pipelining hard to Implement, Extending the MIPS Pipeline to Handle Multicycle Operations, The MIPS R4000 Pipeline, Cross Cutting Issues. ***Text Book-2: C.1 to C.7***	
UNIT – V	**12 Hours**
Memory Hierarchy: *Introduction, Cache Performance, Six basic cache Optimizations, Virtual Memory, Protection and examples of Virtual Memory, Fallacies and Pitfalls.* ***Text Book-2: B.1 to B.6***	

Course Outcome	Description	Bloom's Taxonomy Level
At the end of the course the student will be able to:		
1	**Apply** knowledge of the internal architecture and organization of ARM microprocessors to utilize their components and functionalities.	L3
2	**Apply** the instruction set of the ARM Microprocessor by writing Assembly language programs.	L3
3	**Analyze and compare** the various exception handling techniques.	L4
4	**Examine** the concept of instruction-level parallelism and **analyze** the principles of Pipelining techniques.	L4
5	**Compare** and C**ontrast** memory hierarchy and its impact on computer cost/performance.	L4

Table: Mapping Levels of COs to POs / PSOs														
COs	**Program Outcomes (POs)**												**PSOs**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO 1	3		2										2	
CO 2	3		3		1								2	
CO 3	3	3	1										2	
CO 4	3	3	1										2	
CO 5	3	3	1										2	

3: Substantial (High) 2: Moderate (Medium) 1: Poor (Low)

TEXT BOOKS:

1. William Hohl, "ARM Assembly Language", 2nd Edition, CRC Press, 2009.
2. John L Hennessy, David A Patterson, "Computer Architecture, A Quantitative Approach", 5th Edition, Morgan Kaufmann publishers, 2012.

REFERENCE BOOKS:

1. David A Patterson, John L Hennessy, "Computer Organization and Design", 4th Edition, Morgan Kaufmann publishers, 2010.
2. Steve Furber, "ARM System-on-chip Architecture", 2nd Edition, Pearson Publications, 2000.
3. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill, 2002.

E-Resources:

1. https://www.udemy.com/topic/arm-cortex-m/
2. https://www.edx.org/school/armeducation
3. https://onlinecourses.nptel.ac.in/noc22_cs93/preview

Activity Based Learning (Suggested Activities in Class)

1. Mini project implementation using Assembly Language Programming.
2. Demonstration of solutions to a problem through programming.

****