SYLLABUS
(3rd & 4th Semester)

3rd SEMESTER

Course Title	Applied Mathematics –III for Computer Science and Engineering Stream (AIML, ISE, CSE, CSE-IOT)	**Semester**	III
Course Code	MC201	**CIE**	50
Total No. of Contact Hours	50	**SEE**	50
No. of Contact Hours/week	3:2:0:2	**Total**	100
Credits	4	**Exam. Duration**	3 Hours
Teaching Department	MATHEMATICS		

Course objective is to:

- Have a foundation in the fundamentals of logic, enabling them to apply logical reasoning effectively in their academic and professional pursuits.
- Develop skills in data analysis, problem-solving, and communication, preparing them to apply statistical techniques effectively in various fields such as science and engineering.
- To facilitate students with the knowledge and skills needed to effectively use regression analysis as a tool for understanding data relationships, making predictions, and drawing meaningful conclusions from data.

Module-1	**RBT Levels L1,L2,L3**	**10 Hours**

Mathematical logic:
Fundamentals of Logic: Basic connectives and truth tables, Logical equivalence – The laws of Logic, Logical implication – Rules of Inference. Quantifiers, Validity of Quantifiers.

Module-2	**RBT Levels L1,L2,L3**	**10 Hours**

Probability Distributions:
Review of basic probability theory. Random variables (discrete and continuous), probability mass and density functions. Mathematical expectation, mean and variance. Binomial, Poisson, Exponential and normal distributions.

Module-3	**RBT Levels L1,L2,L3**	**10 Hours**

Joint probability distribution:
Joint Probability distribution for two discrete random variables, expectation, covariance and correlation. Markov Chain: Introduction to Stochastic Process, Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular and absorbing states..

Module-4	**RBT Levels L1,L2,L3**	**10 Hours**

Statistical Methods:

Correlation and Regression-Karl Pearson's coefficient of correlation and rank correlation-problems, Curve Fitting: Curve fitting by the method of least squares- fitting the curves of the form- $y = ax + b$, $y = ax^b$ *and* $y = ax^2 + bx +$ c.

Module-5	**RBT Levels L1,L2,L3**	**10 Hours**

Statistical Inference :
Introduction, sampling distribution, standard error, Type-I and Type-II errors. Testing of hypothesis, levels of significance, confidence limits, sampling of attributes, test of significance for large samples for mean and proportions. Students 't' distribution, Chi-square distribution as a test of goodness of fit.

Suggested Learning Resources:
Textbooks:

1	**Ralph P. Grimaldi, B V Ramana: "Discrete and Combinatorial Mathematics** ", 5th Edition, Pearson Education, 2004.
2	B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers, 44th edition, 2017.
3	Kenneth H. Rosen: "Discrete Mathematics and its Applications", 6th Edition, McGraw Hill, 2007.

Reference Books:

1	Basavaraj S Anami and Venakanna S Madalli: "Discrete Mathematics – A Concept-based approach", Universities Press, 2016
2	D.S. Malik and M.K. Sen: "Discrete Mathematical Structures Theory and Applications, Latest Edition, Thomson, 2004.
3	Ralph P. Grimaldi: "Discrete and Combinatorial Mathematics", 5th Edition, Pearson Education. 2004.
4	Thomas Koshy – Elementary Number Theory with Applications, Academic Press, 2nd edition, 2008.

Web links and Video Lectures (e-Resources):
M1: https://youtu.be/xlUFkMKSB3Y?list=PL0862D1A947252D20
M2: https://youtu.be/_BIKq9Xo_5A
M3: https://youtu.be/Q9HNSfF7Hq4?list=PL3RvMHSY8k8lyQsWfHsFELUjfglQB0Ovn
M4: https://youtu.be/BA1iyzRFGzo?list=PLT3bOBUU3L9h4StzmTBjUnzosZYpCKcKZ
M5: https://youtu.be/OQV8WmUdeIo?list=PLLqEsfz6HOamSu7v9zBZ1IkVcCl2atzWL

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning
Quiz, Group Discussions, Seminar, In class assignments,

Course outcomes:

CO1	Apply concepts of logical reasoning and mathematical proof techniques in proving theorems and statements.
CO2	Explain the basic concepts of probability, random variables, probability distribution
CO3	Apply suitable probability distribution models for the given scenario
CO4	Make use of the analysis to fit a suitable mathematical model for statistical data.
CO5	Compute the confidence intervals for the mean of the population

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2		2								2
CO2	2	2		2								2
CO3	2	2		2								2
CO4	2	2		2								2
CO5	2	2		2								2

High-3, Medium-2, Low-1

Course Title	Digital Design and Computer Organization	**Semester**	III
Course Code	CS202	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:2:0:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

Course objective is to:

1. To understand the basics of Digital System Design.
2. To learn about Combinational and Sequential circuit design.
3. To understand the Basic Structure of Computer and machine instructions.
4. To understand basics of Memory System, Processing unit and Pipeline concept.
5. Identify the applicability of the embedded system for real time use cases.

PREREQUISITES : None

Module-1: Overview and Introduction of Digital System Design	**RBT Levels:L1, L2,L3**	**8 Hours**

Digital System Design: Need for Digital Systems, Summary on Simplification of the Digital design, Analog vs Digital Signals. Basic Theorems and Properties of Boolean Algebra, Boolean functions, Canonical and Standard form, Digital logic gates, The Map Method, Product of sum simplification, Four-Variable Map, Don't-Care Conditions, NAND and NOR Implementation, EX-OR function, Hardware Description Language.

Text Book 1: 2.4, 2.5, 2.6, 2.8, 3.2, 3.3, 3.4, 3.5, 3.6, 3.8, 3.9

Module-2: Combinational and Sequential logic	**RBT Levels:L1,L2,L3**	**8 Hours**

Combinational Logic: Introduction, Combinational Circuits, Binary Adder- Subtractor, Decoders, Encoders, Multiplexers.

Sequential Logic: Introduction, Sequential Circuits, Storage Elements: Latches, Flip-Flops, Registers, Shift Registers, Ripple Counters, Synchronous Counters, Other Counters.

Text Book 1: 4.1, 4.2, 4.5, 4.9, 4.10, 4.11, 5.1, 5.2, 5.3, 5.4, 6.1, 6.2, 6.3, 6.4, 6.5.

Module-3: Basic Structure of Computers and MachineInstructions, Programs	**RBT Levels:L1,L2**	**8 Hours**

Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus structure, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement.

Machine Instructions and Programs: Numbers Arithmetic operations and characters, Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Interrupts.

Textbook 2:1.2,1.3,1.4,1.6, 2.1,2.2,4.1,4.2

Module-4: The Memory System, Basic Processing Unit and Pipelining	**RBT Levels:L1,L2, L3**	**8 Hours**

The Memory System: Some Basic concepts, Internal Organization of memory chips, Structure of large memory, Read only memories, Speed, Size, Cost, Cache memories, Mapping functions, Replacement algorithms, Examples of mapping technique.

Basic Processing Unit: Some fundamental concepts, Execution of Complete instruction.Pipelining: Basic Concepts, Data Hazards, and Instruction Hazards.

Text Book 2: 5.1, 5.2.1,5.2.5, 5.3, 5.4, 5.5, 5.5.1, 5.5.2, 5.5.3, 7.1,7.2,8.1,8.2,8.3

Module-5: Embedded system Architectures and Case study	**RBT Levels:L1,L2,L3**	**8 Hours**

Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, Typical embedded system, Core of the embedded system, General purpose and domain specific processors, Sensors and actuators, Communication interface, Embedded firmware.

Summary on all modules.
Case Study on Automatic Chocolate Vending Machine (ACVM).

Text Book 3:1.2 to 1.6, 2.1, 2.1.1, 2.3, 2.4, 2.5

Follow on Courses: Microcontrollers, ARM, Embedded System, RISC-V, Compiler Design, and IOT.

Topics for Advanced Learners:

1. **RISC-V**

Resources for Advanced Learners:

https://riscv.org/certifications-and-courses/

Suggested Learning Resources:

Textbooks:

1.	M. Morris Mano & Michael D. Ciletti, Digital Design with an Introduction to Verilog HDL, 5th Edition, Pearson Education.
2.	Carl Hamacher, Zvonko Vranesic, SafwatZaky, Computer Organization, 5th Edition, Tata McGraw Hill.
3	Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2nd Edition.

Reference Books:

1.	Fundamentals of Logic Design, Charles H. Roth, 4th Edition, PWS Publishing Company.
2.	Computer Organization and Design: The Hardware/Software Interface Hardcover, David A. Patterson, John L. Hennessy, 6TH Edition, MK Publications

Web links and Video Lectures (e-Resources):

https://onlinecourses.nptel.ac.in/noc21_ee39/preview
https://archive.nptel.ac.in/courses/106/105/106105163/

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Flipped Class Room
2. Presentation

Course outcomes:	
CO1	Ability to simplify Digital Design using various techniques.
CO2	Ability to build Combinational and Sequential circuits.
CO3	Understanding of Basic Structure of Computer and machine instructions.
CO4	Understanding of basic operational concept of computer and pipelining.
CO5	Understanding the basics of Embedded Systems with sample applications.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	2	-	2	3	-	-	2	2	2	-
CO2	3	3	3	3	3	2	-	2	2	-	-	2	2	2	-
CO3	3	3	3	3	3	2	-	2	3	-	-	2	2	2	-
CO4	3	3	3	3	3	2	-	2	2	-	-	2	2	2	-
CO5	3	3	3	3	3	2	-	2	3	-	-	2	2	2	-

High-3, Medium-2, Low-1

Digital Design and Computer Organization Laboratory			
Course Code	**CS203**	**Semester**	**III**
L:T:P:S	**0:0:2:1**	**CIE Marks**	**50**
Hrs. / Week	**2**	**SEE Marks**	**50**
Credits	**1**	**Total Marks**	**100**
Exam Hours	**03**		

Course objectives : At the end of the course, the student will be able to:	
1.	To Simplify the Digital System Design.
2	Flip - Flops and their operations.
3.	Counters and registers using flip-flops.
4.	Synchronous and Asynchronous sequential circuits.
5.	Develop VHDL code for combinational and sequential circuits.

Pgm. No.	List of Experiments / Programs	Hours	COs
Prerequisite Experiments / Programs / Demo			
	NIL		
PART-A			
1	**Title:** Design a Digital System that Connects any one input to output based on selection inputs. **Problem Description**: A Digital system which consists of 8 input lines and at any given time only one input will be connected to output based on select lines. **Method:** The technique utilized here is to simplify the given expression using multiplexer and connect one among multiple input to output signal based on selected inputs. **Theory Reference:** Module 1	2	CO1,CO2
2	**Title:** Simplify the digital design that accepts two/three binary inputs and produces two binary outputs.	2	CO1,CO2
	Problem Description: The technique involves simplifying the digital design to accept two/three binary inputs A&B and return two outputs Sum S and Carry C. **Method:** In this experiment, the truth table of 2-bit and 3-bit adder is converted to logical expression and implementing the simplified expression to verify the truth table. **Theory Reference:** Module 1		

3	**Title:** Design a circuit to store one bit of information using universal gates. **Problem Description:** Realize the working of flip-flops using universal gates. **Method:** In this experiment flip-flops can be built using universal gates in the feed-back path and to demonstrate it can be used as storage elements. **Theory Reference:** Module 2	2	CO1,CO2
4	**Title:** Synchronous counter. **Problem Description:** Given 'n' count, counter is designed to count up from 0 to n up counting. **Method:** In this experiment Counter is designed using appropriate technique which counts from 0 to n digit. **Theory Reference:** Module 2	2	CO1,CO2
5	**Title:** Design asynchronous counter. **Problem Description** Design a counter for the sequence: 0-1-2-3-4-5-6-7-8-9 and display them on 7-segment LED display. **Method:** In this experiment Counter is designed usingappropriate technique which count up from 0 to n (n<=9) and demonstrate on 7-segment display. **Theory Reference:** Module 2	2	CO1,CO2
6	**Title:** Design a code converter circuit. **Problem Description:** In this experiment code conversion is done from Binary to Gray and appropriate technique is used to simplify the logic circuit demonstrate the conversion. **Method:** Given Binary code is converted to Gray code and obtained logic equation is simplified and realized in the hardware. **Theory Reference:** Module 2	2	CO1,CO2
PART-B			
7	**Title:** Simulate a Full Adder in VHDL. **Problem Description:** Develop a program in VHDL to perform full adder operation. **Method:** Design full adder and check the truth table and waveforms generated. **Theory Reference:** Module 2	2	CO1,CO2

8	**Title:** Simulate a 8:1 MUX in VHDL **Problem Description:** Develop a program in VHDL to perform 8:1 multiplexer operation. **Method:** Design 8:1 Multiplexer and check the truth table and waveforms generated. **Theory Reference:** Module 2	2	CO1,CO2
9	**Title:** Simulate a Flip-Flops in VHDL. **Problem Description:** Develop a program in VHDL to check the working of flip-flops (JK/D/T) **Method:** Design Flip-Flops (JK/D/T) and check the operations and waveforms generated. **Theory Reference:** Module 2	2	CO1,CO2
10	**Title:** Simulate a 4-bit binary counter VHDL program. **Problem Description:** Develop a program in VHDL to check 4-bit binary counter. **Method:** Design a 4-bit binary counter and check all combinations generated**.** **Theory Reference:** Module 2	2	CO1,CO2
11	**Title:** Design of a 4-bit Ripple Carry Adder. **Problem Description:** Develop a program in VHDL to perform 4-bit Ripple Carry Adder. **Method:** Design 4-bit Ripple Carry Adder by examining the rippling of carry behaviour. **Theory Reference:** Module 2	2	CO1,CO2
12	**Title:** Design of a 4-bit ALU. **Problem Description:** Develop a program in VHDL to perform 4-bit ALU. **Method:** Design a 4-bit ALU counter and check all combinations generated. **Theory Reference:** Module 2	2	CO1,CO2

PART-C

Beyond Syllabus Virtual Lab Content

Logical Alarm

Suggested Learning Resources:	
Textbooks:	
1	M. Morris Mano & Michael D. Ciletti, Digital Design with an Introduction to Verilog HDL, 5th Edition, Pearson Education.
2	Carl Hamacher, Zvonko Vranesic, SafwatZaky, Computer Organization, 5th Edition, Tata McGraw Hill.
3	Shibu K V, "Introduction to Embedded Systems", Tata McGraw Hill Education, Private Limited, 2nd Edition.
Reference Books:	
1	Fundamentals of Logic Design, Charles H. Roth, 4th Edition, PWS Publishing Company.
2	Computer Organization and Design: The Hardware/Software Interface Hardcover, David A. Patterson, John L. Hennessy, 6TH Edition, MK Publications.
Web links and Video Lectures (e-Resources): https://onlinecourses.nptel.ac.in/noc21_ee39/preview https://archive.nptel.ac.in/courses/106/105/106105163/	
Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning Mini Project	

Course outcomes:	
CO1	Ability to realize Digital System Design and demonstrate using hardware components.
CO2	Ability to demonstrate working of flip-flops and counters.
CO3	Ability to demonstrate working of sequential circuits.
CO4	Proficiency in writing simple programs using VHDL

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	2	-	2	3	-	-	2	2	2	-
CO2	3	3	3	2	2	2	-	2	2	-	-	2	2	2	-
CO3	3	3	3	2	2	2	-	2	3	-	-	2	2	2	-
CO4	3	3	3	2	2	2	-	2	2	-	-	2	2	2	-
CO5	3	3	3	2	2	2	-	2	3	-	-	2	2	2	-

Course Title	Data Structures and Applications	**Semester**	III
Course Code	CS204	**CIE**	50
Total No. of Contact Hours	50	**SEE**	50
No. of Contact Hours/week	3:2:0:2	**Total**	100
Credits	4	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

Course objective is to:

1. To explain fundamentals of Data structures and their applications.
2. To illustrate the representation of various data structures.
3. To learn implementation of all the operations on data structures.
4. To understand the intricacies of various Data Structures thus mapping them to problem solving.

PREREQUISITES : This course requires that the students are familiar with programming language such as C and Python.

Module-1: Overview	**RBT Levels:L1, L2, L3**	**10 Hours**

Data Types to Data Structures: Data Structures, Classifications: Primitive & Non-Primitive (**Stack, Queue, Linked List, Graph, Trees, Hash Table**). Data structure Operations. Review of pointers, Dynamic Memory Allocation Functions with Programming Examples **(Dynamic 1D array, 2D array)**, review of structures, polynomial, Sparse Matrices, string pattern matching (nfind, KMP)

Text Book 1: 1.2,2.1-2.4,2.5.1-2.5.3,2.7.3

Module-2: Stacks and Queues	**RBT Levels:L1, L2, L3**	**10 Hours**

Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays. Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. Multiple stacks, Tower of Hanoi**. Queues:** Definition, Array Representation, Queue Operations,Priority Queues, Circular Queues, Circular queues using Dynamic arrays

Text book 1:3.1,3.2,3.3,3.4,3.6,3.7, 9.1-9.2

Module-3: Linked List: its types and operations	**RBT Levels:L1, L2, L3**	**10 Hours**

Linked Lists: Definition, Representation of linked lists in Memory, Linked list operations: Traversing, Searching, Insertion, and Deletion, Doubly Linked lists Circular linked lists and header linked operations: Traversing, Searching, Insertion, and Deletion, polynomial using header linked list.

Text book 1:4.1,4.2,4.3,4.4,4.5,4.8

Module-4: Trees data structure	**RBT Levels:L1, L2, L3**	**10 Hours**

Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals – In-order, post-order, pre-order; Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching. AVL Trees and Red Black Trees construction, Selection Trees, Forest.

Text book 1:5.1-5.3,5.7-5.9,10.2,10.3

Module-5: Graphs, Hashing and Case Studies	**RBT Levels:L1, L2, L3**	**10 Hours**

Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph operations,
Hashing: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing\

Summarization of all modules.
Case Studies:

a. **Design a music playlist system:** The functionality of a playlist needs to be implemented, i.e., adding a song to the playlist, playing the next song, playing the previous song, switching to a song, display of songs based on its genre type etc.

a. **Efficiently manage table reservations for a restaurant**: Managing table reservations for a restaurant efficiently involves handling various aspects such as availability of tables, booking requests, and ensuring a smooth flow of operations.

Text book1: 6.1,6.2,8.1,8.2,8.3

Follow on Courses: Design & Analysis of Algorithms, Operating Systems

Resources for Advanced Learners:

Data Structures and Algorithms Made Easy – Data Structures & Algorithmic Puzzles Author: Narasimha Karumanchi (MTech. IIT Bombay, Founder- CareerMonk.com) 5th Edition

Suggested Learning Resources:

Textbooks:

1.	Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2.	Yedidyah te, Moshe J. Augenstein and Aaron M. Tenenbaum, Data Structures using C and C++, 2nd edition 2023.
Reference Books:	
1.	Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
2.	Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications,2nd Ed, McGraw Hill, 2013
3.	Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.

Web links and Video Lectures (e-Resources):
1. http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS35.html
2. https://nptel.ac.in/courses/106/105/106105171/
3. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.htm

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Real world problem solving using group discussion.
2. Back/Forward stacks on browsers.
3. Undo/Redo stacks in Excel or Word.

Course outcomes:	
CO1	Ability to explain different data structures and their applications.
CO2	Ability to apply Stacks and Queue data structures to solve the given problems.
CO3	Ability to use the concept of linked list in problem solving.
CO4	Ability to explain the advanced Data Structures concepts such as Hashing Techniques and augmentednbinary search trees.
CO5	Ability to develop solutions using trees and graphs to model the real-world problem.

CO-PO Mapping															
CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	2	2	-	-	-	1	1	-	-	3	3	2
CO2	3	2	3	2	2	-	-	-	1	1	-	-	3	3	2
CO3	3	2	3	2	2	-	-	-	1	1	-	-	3	3	2
CO4	3	2	3	2	2	-	-	-	1	1	-	-	3	3	2
CO5	3	2	3	2	2	-	-	-	1	1	-	-	3	3	2

High-3, Medium-2, Low-1

Data Structures Laboratory			
Course Code	**CS205**	**Semester**	**III**
L:T:P:S	**0:0:2:1**	**CIE Marks**	**50**
Hrs. / Week	**2**	**SEE Marks**	**50**
Credits	**1**	**Total Marks**	**100**
Exam Hours	**03**		

Course objectives : This laboratory course enables students to	
1.	Get practical experience in design, develop, implement Linear data structures and their applications.
2	Get practical experience to design, develop and implement Non-Linear data structures and their applications.

Pgm. No.	**List of Experiments / Programs**	**Hours**	**COs**
	Prerequisite Experiments / Programs / Demo		
This course requires that the students are familiar with programming language such as C(23POP103/203).			
	PART-A		
1	**Title:** Write a program in C to create a data structure which can handle varying number of entries and sort it., **Problem Description:** Need to create a data structure which makes use of memory dynamically. The sorting can be done using any technique. **Method:** The method utilized here is the use of dynamic memory allocation functions and define a function to perform selection sort. **Theory Reference:** Module 1	2	CO2,CO3
2	**Title**: Design, develop, and implement a data structure of integers that follows the LIFO principle and demonstrates its operations. **Problem Description:** The program creates the data structure and performs operations like inserting an element, deleting an element, and displaying the status of the data structure.	2	CO2,CO3

	Method: In this program, create the data structure and support the program with appropriate functions for each of the above operations. Demonstrate Overflow and Underflow situations and support the program with appropriate functions for each of the above operations. When overflow occurs increase the size of the datastructure by array doubling technique. **Theory Reference:** Module 1		
3	**Title:** Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression and Evaluate the converted postfix expression. **Problem Description:** The problem involves implementing a function that convert the Infix Expression to Postfix Expression anda function that evaluates the converted expression and prints the result. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands. **Method:** In this program, Infix expression is given as an input andthe function for converting infix to postfix expression. After obtaining the postfix expression for each alphanumeric operand's values are given at the runtime and display the resultant value. Choose the best data structure for these operations. **Theory Reference:** Module 1	2	CO2,CO3
4	**Title:** Design, develop, and implement an efficient use of memorydata structure of Integers that follows the FIFO principle anddemonstrates its operations. **Problem Description:** The program creates the data structure and performs operations like inserting an element, deleting an element,and displaying the status of the data structure. **Method:** In this program, create the data structure and support the program with appropriate functions for each of the above operations. Demonstrate Overflow and Underflow situations and support the program with appropriate functions for each of the above operations. **Theory Reference:** Module 2	2	CO2,CO3
5	**Title:** Design, Develop and Implement a menu driven Program in C for handling change of branch. **Problem Description:** After 1st year, students can apply forchange of branch and are generally selected based on their CGPA. Assume that among the selected students, some may change their mind. Some existing students also may choose to leave and join another department. Display the complete	2	CO2,CO3

	student data with regular and change of branch students, Limit your program to only your department. **Method: M**ake use of arrays or linked lists. **Theory Reference:** Module 2		
	PART-B		
6	**Title:** Design, Develop and Implement a menu driven Programin C for traversing a tree and search a given item. **Problem Description:** Create a hierarchical data structure and perform efficient insertion and search. Given a set of data items, Create a hierarchical data structure of N nodes, with certain order. Implement all Traversal of the created data structure and output the nodes visited. Iii) Perform efficient Search for an element (KEY) in the created data structure and report the appropriate message. **Method:** Make use of binary search tree. You can use recursion or iterative techniques. **Theory Reference:** Module 3, Binary Search Tree.	2	CO1,CO2,CO3
7	**Title:** Develop a C program to implement height balanced tree **Problem Description:** Given an imbalanced tree, restructure it as a balanced tree. **Method:** Make use of AVL tree. Ensure that the program performs proper rotations such as Left-Left, LR, RR and RL to create a height balanced binary search tree and display the tree created as a output . **Theory Reference:** Module 4	2	CO1,CO2,CO3
8	**Title:** Design, Develop and Implement a Program in C to createN router connections and display the routers by removal of which will disconnect the network completely. **Problem Description:** create a data structure to represent the routers and its connections between the routers and display the router numbers on removal which it disconnects the network completely. **Method:** choose the best data structure to represent the routers and a function to find the articulation point. **Theory Reference:** Module 5	2	CO1,CO2,CO3
9	**Title:** Design and develop a Program in C that uses Hash function H: K ->L as H(K)=K mod m and implement hashing	2	CO1,CO2,CO3

	technique to map a given key K to the address space L. Resolve the collision (if any). **Problem Description:** Given a set K of Keys (4-digit) which uniquely determines the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. **Method:** Ensure that the program generates the 4-digit key randomly and generates the L using hash function, demonstrates if there is a collision and its resolution by using linear probing. **Theory Reference:** Module 5		
10	**Title:** Design and develop a Program in C to implement Merkle Tree typically used in Blockchain. **Problem Description:** Generate a Merkle tree to store a given block of transactions in a tamper-proof manner. **Method:** In a Merkle Tree, each leaf node is labelled with the hash value of a data block and each non-leaf node is labelled with the hash value of its child nodes labels. Ensure that the program should uses SHA-256 hash function to hash transaction data continuously till the Merkle root is obtained. **Theory Reference:** Module 5	2	CO1,CO2,CO3
PART-C **Beyond Syllabus Virtual Lab Content**https://ds1-iiith.vlabs.ac.in/exp/linked-list/doubly-linked-list/dllexercise.html			

Suggested Learning Resources: **Textbooks:**	
1	Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2	Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
Reference Books:	
1	Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
2	Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications,2nd Ed, McGraw Hill, 2013
3	Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.
Web links and Video Lectures (e-Resources): 1. http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS35.html 2. https://nptel.ac.in/courses/106/105/106105171/ 3. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.htm	

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Real world problem solving using group discussion.
2. Back/Forward stacks on browsers.
3. Undo/Redo stacks in Excel or

Course outcomes:	
CO1	Ability to analyze and compare various linear and non-linear data structures
CO2	Ability to Code, debug and demonstrate the working nature of different types of data structures and their applications.
CO3	Ability to choose the appropriate data structure for solving real world problems

CO-PO Mapping															
CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO 3
CO1	3	3	2	2	3	-	-	-	-	-	-	1	3	3	2
CO2	3	3	2	2	3	-	-	-	-	-	-	1	3	3	2
CO3	3	3	2	2	3	-	-	-	-	-	-	1	3	3	2

Course Title	Object Oriented Programming with Java	**Semester**	III
Course Code	CS206	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:0:2:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

PREREQUISITES: This course requires that the students are familiar with programming language such as C(23POP103/203) and python(23PLB105/205).

Course objective is to:

1. To understand Object Oriented Programming Features of JAVA.
2. To learn the concept classes and their interrelationships and how to explode them application development.
3. To develop programs which enable concurrent and parallel processing.
4. To understand and implement error processing in java.
5. To learn how to develop a java application using a standard framework.

Module-1: Overview of Object-oriented Programming and Java Basics	**RBT Level: L1**	**8 Hours**

Object oriented Programming: Need for OOP paradigm, summary of OOP concepts. **Java Basics:** Java buzzwords, data types, variables, scope and lifetime of variables, arrays, expressions, control statements (for each), type conversion and casting, simple java program.

Text Book:1, Chapters 2,3

Module-2: Classes, Objects and Methods	**RBT Levels: L1, L2**	**8 Hours**

Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection.

Methods and Classes: Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, Understanding static, Introducing final, Introducing Nested and Inner Classes.

Text book: 1, Chapters 6,7

Module-3: Inheritance and Interfaces	**RBT Levels: L1, L2**	**8 Hours**

Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Local Variable Type Inference and Inheritance.

Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.

Text book:1, Chapter 8, 9.4-9.6, 9.7

Module-4: Exception handling and Multithreading	RBT Levels: L1, L2	8 Hours

Exception handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions.

Multithreading: Thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication. Enumerations, autoboxing.

Text book: 1, Chapter 10, 11.1,11.3, 11,6, 11.7-11.8,12.1, 12.3

Module-5: Swings, Exploring Swing and Case Studies	RBT Levels: L1, L2, L3	8 Hours

Introducing Swing: The Origins of Swing, Swing Is Built on the AWT, Two Key Swing Features, The MVC Connection, Components and Containers, The Swing Packages, A Simple Swing Application. **Exploring Swing:** JLabel and ImageIcon, JTextField, The Swing Buttons, JTabbedPane, JScrollPane, JList, JComboBox, Trees, JTable. **Case Studies: Hangman Game, Tic Tac Toe Game.**

Summarization of all modules.
Text book: 1, Chapter 32, 33

Follow on Courses: Advanced Java Programming, Object-Oriented Analysis and Design

Resources for Advanced Learners

1. Advanced Java Tutorial | Mastery in Java Programming - GeeksforGeeks

Practice Component of IPCC	L:T:P (0:0:2)

Programming Experiments (Suggested and are not limited to)

Program 1:

Title: Write a program in java to pass argument and count how many parameters are passed.

Problem Description: The program counts the number of parameters passed at the command prompt and displays them along with their occurrence.

Method: The method utilized here is to access the command line arguments using the args array in the main method. The length of this array is then used to determine the number of parameters passed.

Theory Reference: Module 1

Program 2:

Title: Write a java program to create and display the student details.

Problem Description: The problem involves creating a Java class named Student, which contains variables for USN, Name, Branch, and Phone. The task further requires writing a Java program to instantiate 'n' Student objects and printing their details with appropriate headings.

Method: In this program, the 'Student' class encapsulates the details of a student, and the main program creates an array of Student objects, initializes each object with sample data, and then prints out the details of each student.

Theory Reference: Module 2

Program 3:

Title: Write a java program to implement staff hierarchy**.**

Problem Description: The problem involves designing a superclass named Staff with attributes such as: StaffId, Name, Phone, and Salary. Additionally, three subclasses: Teaching, Technical, and Contract-are to be created with specific attributes for each category. The task requires writing a Java program to read and display at least three staff objects from all three categories.

Method: In this example, the Staff superclass encapsulates common attributes shared by all staff members, while the Teaching, Technical, and Contract subclasses define specific attributes for each category of staff. The main program creates and displays at least three staff objects from each category.

Theory Reference: Module 3

Program 4:

Title: Write a java program to implement Box class application to depict constructor overloading.

Problem Description: Given three classes called A, B, and C, C can be a subclass of B, which is a subclass of A. When this type of situation occurs, each subclass inherits all of the traits found in all of its super classes. methods with different parameters.

Method: In this program, C inherits all aspects of B and A. To see how a multilevel hierarchy can be useful, consider the following program. In it, the subclass BoxWeight is used as a superclass to create the subclass called Shipment. Shipment inherits all of the traits of BoxWeight and Box, and adds a field called cost, which holds the cost of shipping such a parcel.

Theory Reference: Module 3

Program 5:

Title: Write a java program to solve Tower of Hanoi Problem using Stack.

Problem Description: Move all the disks stacked on the first tower over to the last tower using a helper tower in the middle. While moving the disks, certain rules must be followed. Only one disk can be moved. A larger disk cannot be placed on a smaller disk.

Method: In this program, Solving the Tower of Hanoi problem using a stack is an elegant and efficient approach. The Tower of Hanoi problem involves three rods and a number of disks of different sizes that can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, with the smallest disk at the top, and the objective is to move the entire stack to another.

Theory Reference: Module 3

Program 6:

Title: "Write a Java Program to calculate area and perimeter of variety of shapes (circle and triangle)"

Problem Description: Develop a JAVA program to create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape.

Method: Ensure that the program is well-structured, follows object-oriented principles, and provides clear and concise output demonstrating the functionality of each class and method.

Theory Reference: Module 3 **Page no**:157

Program 7:

Title: "Java Program: Resizable Interface for Object Resizing with Rectangle Implementation".

Problem Description: Develop a JAVA program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.

Method: Ensure that the program follows proper object-oriented principles, such as encapsulation and abstraction, and provides clear and concise output demonstrating the resizing functionality.

Theory Reference: Module 3

Program 8:

Title: "Java Program: Custom Exception Handling for DivisionByZero and Arithmetic Exceptions".

Problem Description: Develop a Java program that demonstrates the handling of custom exceptions, specifically for DivisionByZero and Arithmetic exceptions. You are required to utilize try-catch blocks along with throw statements to handle these exceptions gracefully.

Method: Ensure that the program demonstrates the proper usage of try-catch blocks, throw statements, and custom exception handling for DivisionByZero and Arithmetic exceptions.

Theory Reference: Module 4

Program 9:

Title: "Write a Java program to generate random numbers using multiple threads.

Problem Description: Develop a Java program that implements a multi-threaded application with three threads. Each thread has a specific task as described as follows: First Thread (Random Number Generator): This thread generates a random integer every 1 second. Second Thread (Square Computation): This thread receives the random integer generated by the first thread and computes its square. After computing the square, it prints the result. Third Thread (Cube Computation): This thread receives the random integer generated by the first thread and computes its cube. After computing the cube, it prints the result.

Method: Program should demonstrate the multi-threading capabilities of Java and showcase the asynchronous computation of squares and cubes. Additionally, it should handle synchronization and data sharing effectively to prevent race conditions and ensure thread safety.

Theory Reference: Module 4

Program 10:

Title: "Java Swing Program: Creating Buttons with JFrame Inheritance".

Problem Description: Develop a Java program using Swing to create a button and add it to a JFrame object inside the main method. Additionally, you should inherit the JFrame class without explicitly creating an instance of the JFrame class.

Method: Program should demonstrate the creation of a Swing button and its addition to a JFrame object, while also showcasing inheritance of the JFrame class.

Theory Reference: Module 5

Open ended experiments leading to guided projects:
Mobile App Development

Suggested Learning Resources:
Textbooks:

1.	Java: The Complete Reference, Twelfth Edition, by Herbert Schildt, November 2021, McGraw-Hill, ISBN: 9781260463422.

Reference Books:

1.	Programming with Java, 6th Edition, by E Balagurusamy, Mar-2019, McGraw Hill Education, ISBN:9789353162337.
2.	Thinking in Java, Fourth Edition, by Bruce Eckel, Prentice Hall, 2006 (https://sd.blackball.lv/library/thinking_in_java_4th_edition.pdf)

Web links and Video Lectures (e-Resources):

1. Learn Java - A Beginners Guide for 2024 - GeeksforGeeks
2. **Java Tutorial: https://www.javatpoint.com/java-tutorial**
3. **https://www.youtube.com/watch?v=GoXwIVyNvX0**
4. https://www.youtube.com/watch?v=bm0OyhwFDuY&list=PLsyeobzWxl7pe_IiTfNyr55kwJPWbgxB5

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Demonstration of online IDEs like geeksforgeeks, jdoodle or any other Tools.
2. Mini Project

Course outcomes:	
CO1	Ability to demonstrate proficiency in writing simple programs involving branching and looping structures and design class involving data members and methods for the given scenario.
CO2	Ability to apply the concepts of inheritance and interface in solving real world problems.
CO3	Ability to use the concept of multithreading and exception handling in solving complex problem.
CO4	Ability to use the concept of swings in solving the real-world problems.

CO-PO Mapping

CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	1	1	-	-	-	2	-	-	-	3	3	3
CO2	3	2	3	1	1	-	-	-	2	-	-	-	3	3	3
CO3	3	2	3	1	1	-	-	-	2	-	-	-	3	3	3
CO4	3	2	3	1	1	-	-	-	2	-	-	-	3	3	3

High-3, Medium-2, Low-1

Course Title	Operating Systems	**Semester**	III
Course Code	CS221	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:1:1:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

PREREQUISITES: C Programming Course.

Course objective is to:

1. To introduce the concepts, roles, and functions of traditional and modern operating systems.
2. To explain the concept of process and its management which includes process scheduling.
3. To understand the problems related to concurrency, different synchronization mechanisms and deadlock handling.
4. To Illustrate the various memory management and file management techniques.
5. To understand different APIs/Commands related to processor, memory, storage, and file system management.

Module-1: Overview and structure of Operating System	**RBT Levels:L1,L2**	**8 Hours**

Overview Operating Systems: Definition - Operating System, computing environment and nature of computations, classes of operating systems, Efficiency, system performance and user services, Batch processing systems, multiprogramming systems, time sharing systems, real time OS, distributed OS, Modern OS.

Structure of Operating Systems: operation of an OS, structure of an OS, layered design of OS, Virtual machine OS.

Text book 1: Chapter 3.1 to 3.9, 4.1 to 4.4

Module-2: Process, Threads and Process Synchronization	**RBT Levels:L1,L2,L3**	**8 Hours**

Process and Threads: Processes and programs, Implementing process, case study: Threads, processes in UNIX.

Process Synchronization: Race conditions, Critical section, control synchronization and invisible operations, synchronization approaches, structure of concurrent systems, classic approach synchronization problems, semaphores, monitors, case study: process synchronization in UNIX.

Textbook 1: Chapter 5.1 to 5.3, 5.4.1, 6.1 to 6.7, 6.9, 6.10, 6.11.2.

Module-3: Scheduling and Deadlocks	**RBT Levels:L1,L2,L3**	**8 Hours**

Scheduling: Scheduling Terminology and concepts, non-pre-emptive and pre-emptive scheduling policies, scheduling in practice, Case study: scheduling in UNIX.

Deadlocks: what is deadlock? Deadlocks in resource allocation, Handling Deadlocks, Deadlock Detection and Resolution, Deadlock prevention, deadlock avoidance, case study: Deadlock handling in UNIX.

Textbook 1: Chapter 7.1 to 7.4, 7.6.1, 8.1 to 8.6, 8.8.1.

Module-4: Memory Management and Virtual memory	**RBT Levels:L1,L2,L3**	**8 Hours**

Memory Management: Managing the memory hierarchy, static and dynamic allocation, execution of program, memory allocation to process, contiguous and non-contiguous memory allocation, paging, segmentation, segmentation with paging.

Virtual memory: Basics, demand paging, the virtual memory manager, page replacement policies, Case study: UNIX virtual memory

Textbook 1: Chapter 11.1 to 11.4, 11.6 to 11.10

Module-5: File Systems and Mass Storage Structures	**RBT Levels:L1,L2,L3**	**8 Hours**

File Systems: overview of File Processing. Files and file operation, fundamental file organization and access methods, directories, case study: Unix File system.

Mass Storage Structures: Overview; Disk Structure, Scheduling, Management.

Textbook 1: Chapter 13.1 to 13.4, 13.14.1 Textbook 2: chapter 12.1,12.2,12.4,12.5.

Follow-on Courses: System Software Compiler Design, Embedded Systems and Cloud Computing.

Resources for Advanced Learners

An Introduction to Operating Systems Concepts and Practice, PCP Bhatt, 2013 https://www.amazon.in/INTRODUCTION-OPERATING-SYSTEMS-CONCEPTS-PRACTICE-dp-9387472876/dp/9387472876/ref=dp_ob_title_bk

Practice Component of IPCC **L:T:P (0:0:2)**

Programming Experiments (Suggested and are not limited to)

Program 1:

Title: Demonstrate operating systems Installation and Basic Unix/Linux commands like ... pwd, cd, ls ls-l , rm, mv, cp, chmod, mkdir rmdir, find, file, cat, wc, sort, grep etc…

Problem Description: To learn the OS installation Process and the purpose of basic Commands in Unix/Linux.

Method: Follow the procedure for installation, open terminal and execute basic commands by understanding the purpose of each command.

Theory Reference: Module 1

Program 2:

Title: Write a C/C++/Python code to implement process creation.

Problem Description: The students can create child process and recognize the Id's of both parent and child process.

Method: Follow the method to create a child process by a parent process in an operating system using a fork () system call. A child process may also be known as a subprocess or a subtask. A child process is created as a copy of its parent process. The child process inherits most of its attributes.

Theory Reference: Module 1

Program 3:

Title: Write a C/C++/Python code to implementing the Process system calls for inter process communication.

Problem Description: To learn how OS System calls allow programs to create and terminate processes, as well as manage inter-process communication.

Method: Follow the procedure for using various types of system calls in operating systems, including process control system calls, file management system calls, device management system calls, information maintenance system calls, and communication system calls.

Theory Reference: Module 1

Program 4:

Title: Write a C/C++/Python code to Implementing I/O system calls to demonstrate file operations.

Problem Description: To learn the OS I/O system calls open , close , read , and write are commonly used in programming to interact with files and perform input/output operations.

Method: Follow the procedure

1. open(name, mode) – Used to open a file name in the mode (read or write) specified.0 isfor opening in read mode, 1 for writing and 2 for both.
2. close(fd) – Close an opened file.
3. read(fd, buffer, n_to_read) – Read data from a file.
4. write(fd, buffer, n_to_write) - write data from to a file.

Theory Reference: Module 1

Program 5:

Title: Write a C/C++/Python code to Implementing IPC using message queues using socket program.

Problem Description: To learn IPC where an application sends a message to an operating system process. The operating system sends the message to a designated IPC mechanism, which handles the message and sends a response back to the application. IPC mechanisms can be found in the kernel or the user.

Method: Follow the procedure to create IPC using Unix domain socket through UDS or IPC socket (inter-process communication socket) is a data communications endpoint for exchanging data between processes executing on the same host operating system.

Theory Reference: Module 2

Program 6:

Title: Write a C/C++/Python code to implement Bankers Algorithm for Deadlock avoidance in resources allocation process.

Problem Description: To learn how to prevent the situation where two or more processes are unable to proceed because each is waiting for one of the others to release a resource.

Method: Follow the procedure for deadlock avoidance by eliminating any of the four necessary conditions, which are mutual exclusion, hold and wait, no preemption, and circular wait.

Theory Reference: Module 3

Program 7:

Title: Write a C/C++/Python code for FCFS and SJF Scheduling to allocate the jobs to CPU for processing jobs.

Problem Description: To learn the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process based on a particular strategy. Process scheduling is an essential part of a Multiprogramming operating system.

Method: Follow the procedure for scheduling the process into ready, waiting, and operating states via process scheduling algorithm FCFS (allocates the process that arrived first to the CPU) and SJF(allocates the process that has the shortest burst time to the CPU).

Theory Reference: Module 2

Program 8:

Title: Write a C/C++/Python code to implement round robin scheduling algorithm to allocate the computing resource uniformly to all the jobs running on the system without delay.

Problem Description: To learn the activity of the process manager that handles the removal of the running process from the CPU and the selection of another process that is allocated to CPU based on a particular strategy. Process scheduling is an essential part of a Multiprogramming operating system.

Method: Follow the procedure for scheduling the process into ready, waiting, and operating states via process scheduling algorithm Round Robin (allocates the process in a circular fashion on regular time interval to the CPU).

Theory Reference: Module 2

Program 9:

Title: Write a C/C++/Python code to implement FIFO Page Replacement Algorithm to efficiently use the frames.

Problem Description: To learn page replacement the process in which a page from the main memory is replaced with a page from the secondary memory.

Method: Follow the procedure for page replacement using FIFO Page Replacement Algorithm(replaces the page that has come first.

Theory Reference: Module 4

Program 10:

Title: Write a C/C++/Python code to implement LRU and LFU Page Replacement Algorithm to load the pages to frames efficiently.

Problem Description: To learn page replacement the process in which a page from the main memory is replaced with a page from the secondary memory.

Method: Follow the procedure for page replacement using LRU (replace the page that has not been used recently) and LFU(replace the page that will not be used in the near future) Page Replacement Algorithm.

Theory Reference: Module 4

Program 11:

Title: Write a C/C++/Python code to implement first fit, best fit and worst fit algorithms for efficient memory management.

Problem Description: To learn memory management the process of controlling and coordinating a computer's main memory. It ensures that blocks of memory space are properly managed and allocated so the operating system (OS)

Method: Follow the procedure for searching through the list of free blocks of memory, starting from the beginning of the list and implement First Fit(searches for the first free partition that is large enough to accommodate the process) and Best Fit(searches through the list of free blocks of memory to find the block that is closest in size to the memory request from the process) algorithm.

Theory Reference: Module 4

Open ended experiments leading to guided projects:

1. Develop a lightweight Operating System

Suggested Learning Resources:

Textbooks:

1.	Dhananjay M Dhamdhere, Operating systems: a Concept- Based Approach, McGraw hill, 3rd Edition 2012.
2.	Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, John Wiley & Sons, Inc., 9th Edition, 2012, ISBN 9781118063330.
3.	Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH, 2017.
Reference Books:	
1.	Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2.	William Stallings, "Operating Systems: Internals and Design Principles", Eighth Edition, Prentice Hall, 2015.
3.	Unix Shell Programming, Yashwant Kanetkar.
4.	Introduction to UNIX by M G Venkatesh Murthy.
5.	Practical Linux System Administration. by Kenneth Hess.
6.	Beginning Linux System Administration. by Sander van Vugt.

Web links and Video Lectures (e-Resources):

https://www.w3schools.in/operating-system/linux-operating-system

https://www.tutorialspoint.com/unix/unix-getting-started.htm

https://www.javatpoint.com/unix-operating-system

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1.Setting up a VMware Workstation2.Installing

a VMware Guest OS

3.Running LINUX Operating System

Course outcomes:	
CO1	Identify the roles and functions of traditional and modern operating systems.
CO2	Analyse the various techniques for process management and process scheduling
CO3	Evaluate the problems related to concurrency, different synchronization mechanisms and deadlock handling.
CO4	Apply various technique for memory management
CO5	Explain file and secondary storage management strategies

CO-PO Mapping															
CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	2	3	2	2
CO2	3	2	3	-	-	-	-	-	-	-	-	2	3	2	2
CO3	3	2	3	-	2	-	-	-	-	-	-	2	3	2	2
CO4	3	2	3	1	2	-	-	-	1	1	-	2	3	2	2
CO5	3	2	3	-	-	-	-	-	-	-	-	2	3	2	2

High-3, Medium-2, Low-1

Course Title	Unix Programming	**Semester**	5
Course Code	CS222	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:1:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE / ISE		

Course objective is to:

- To help the students to understand effective use of Unix concepts, commands and terminology.
- Identify, access, and evaluate UNIX file system
- Explain the fundamental design of the unix operating system
- Familiarize with the systems calls provided in the unix environment
- Design and build an application/service over the unix operating system

PREREQUISITES: A basic knowledge of computers, programming language like C, operating system.

Module-1 Introduction, Unix Files	**RBT Level – 1, 2**	**8 Hours**

Introduction: Unix Components/Architecture. Features of Unix. The UNIX Environment and UNIX Structure, Posix and Single Unix specification. General features of Unix commands/ command structure. Command arguments and options. Basic Unix commands such as echo, printf, ls, who, date, passwd, cal, Combining commands. Meaning of Internal and external commands. The type command: knowing the type of a command and locating it. The root login. Becoming the uper user: su command.
Unix files: Naming files. Basic file types/categories. Organization of files. Hidden files. Standard directories. Parent-child relationship. The home directory and the HOME variable. Reaching required files- the PATH variable, manipulating the PATH, Relative and absolute pathnames. Directory commands – pwd, cd, mkdir, rmdir commands. The dot (.) and double dots (..) notations to represent present and parent directories and their usage in relative path names. File related commands – cat, mv, rm, cp, wc and od commands.
Text Book1: Chapter-1, 2, 3, 4, 5

Module-2 Attributes, Commands, Shell Programming	**RBT Level – 1, 2, 3**	**8 Hours**

File attributes and permissions: The ls command with options. Changing file permissions: the relative and absolute permissions changing methods. Recursively changing file permissions. Directory permissions.
The shells interpretive cycle: Wild cards. Removing the special meanings of wild cards. Three standard files and redirection.
Connecting commands: Pipe. Basic and Extended regular expressions. The grep, egrep. Typical examples involving different regular expressions.
Shell programming: Ordinary and environment variables. The. profile. Read and read-only commands. Command line arguments. exit and exit status of a command. Logical operators for conditional execution. The test command and its shortcut. The if, while, for and case control statements. The set and shift commands and handling positional parameters. The here (<<) document and trap command. Simple shell program examples.
Text Book1: Chapter-6,8,13,14

Module-3 Unix Standardization, File I/O, Directories, Unix Process	**RBT Level – 1, 2, 3**	**8 Hours**

Unix Standardization and Implementations: Introduction, Unix Standardization, UNIX System Implementation.
File I/O: Introduction, File Description, open, create, read, write, close, fcntl functions.
Files and Dictionaries: mkdir and rmdir functions, reading dictionaries, chdir, fchdir and getcwd functions. Device Special files.
The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions.
Text Book 2: 2,3,4,7.

Module-4 Process Control, IPC	**RBT Level – 1, 2, 3**	**8 Hours**

Process Control: Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions.
Overview of IPC: Methods, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores.
Shared Memory, Client-Server Properties, Passing File Descriptors, An Open Server-Version 1.
Text Book2: Chapter 8, 15,17

Module-5 Signals and Daemon Process	**RBT Level – 1, 2, 3**	**8 Hours**

Signals and Daemon Processes: Introduction, Signal Concepts, Signal Functions, SIGCLD Semantics, Kill and Raise functions, Alarm and Pause Functions, Signal Sets, sigprocmask Function, sigpending function, sigaction function, sigsetjmp and siglongjmp functions, sigsuspend function, abort function, system function, sleep, nanosleep and clock_nanosleep functions, sigqueue functions, job-control signals, signal names and numbers.
Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model.
Text Book 2: Chapter 10, 13

Follow on Courses: Operating Systems, Computer Systems, Computer Networks.

Resources for Advanced Learners:
https://www.youtube.com/watch?v=ffYUfAqEamY
https://www.youtube.com/watch?v=Q05NZiYFcD

Suggested Learning Resources:
Textbooks:

T1.	Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill
T2.	W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, Pearson Education, 2005
Reference Books:	
R1.	Unix System Programming Using C++ - Terrence Chan, PHI, 1999.
R2.	M.G. Venkatesh Murthy: UNIX & Shell Programming, Pearson Education.
R3.	Richard Blum, Christine Brenham: Linux Command Line and Shell Scripting Bible, 2ndEdition, Wiley, 2014.

Web links and Video Lectures (e-Resources):
https://www.youtube.com/watch?v=8GdT53KDIyY
https://www.youtube.com/watch?app=desktop&v=3Pga3y7rCgo

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1) Programming assignment -1 (Shell level) - 10 marks
2) Programming assignment -2 (API level) - 15 marks.

Course outcomes: Ability to	
CO1	Demonstrate the basics of Unix concepts and commands
CO2	Demonstrate the UNIX file system.
CO3	Apply commands to reflect changes in file system.
CO4	Demonstrate IPC and process management.
CO5	Develop an application/service over a Unix system

	CO-PO Mapping														
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	-	2	1	-	-	1	-	1	-	-	3	-	-
CO2	3	2	1	1	1	-	-	1	1	1	-	1	3	-	-
CO3	3	3	2	2	1	-	-	1	1	1	-	1	3	-	-
CO4	3	3	2	2	1	-	-	1	1	1	-	1	3	-	-
CO5	3	3	3	3	3			1	1	1	-	-	3	-	-

High-3, Medium-2, Low-1

Course Title	Biology for Engineers	**Semester**	III
Course Code	BG207	**CIE**	50
Total No. of Contact Hours	30	**SEE**	50
No. of Contact Hours/Week	2:0:0:1	**Total**	100
Credits	2	**Exam. Duration**	3 Hours
Teaching Dept.	Basic Sciences		

Course objective is to enable students:

- Acquire an understanding on basic modern biological concepts with an emphasis on how bioprocesses are analogous to engineering field, as a multidisciplinary field.
- Understand basic engineering principles imminently run physiological processes particularly about engineering designs and solutions that are arrived citing body functional examples.
- Explain aspects that many bio-solutions could be foundational to design, develop better processes, products and useful to achieve quality of life.

Module-1	**RBT Level L1, L2**	**6 Hours**
Biomimetics: Biology for Engineers, Body Fluid: Blood – Mechanics of heart, Blood pressure, Life molecules: Water, Carbohydrates, Proteins, Lipids and Nucleic acids, Biomimetics: Bio-processes- engineering analogies.		
Module-2	**RBT Level L1, L2**	**6 Hours**
Bioenergy: Unit of life: Human and Plant cell, Metabolism: Enzymes as Bio-catalysts and physiological entities, Anabolism – Bioenergy from Sun-Photosynthesis, catabolism.		
Module-3	**RBT Level L1, L2**	**6 Hours**
Biomechanics (Human body Movement Mechanics) Normal Human Movement: Force-vector of body; Movement Angles; Muscle contraction-relaxation; Posture – Static & Dynamic; Ideal and abnormal posture, Practical: Stepping- Lifting- Sit-Stand.		
Module-4	**RBT Level L1, L2**	**6 Hours**
Bioelectronics Brain & Computer: Senso-neural networks, Biosensors and IoT as applied to biology, Bionic Eye: Mechanism of Vision, Electronic Nose: Bio-olfactory mechanisms (Science of smell), Impulses: Cardiac and Nerve, Biological Clock and Circadian rhythm/		
Module-5	**RBT Level L1, L2,L3**	**6 Hours**
Biopharma: Metabolic syndromes, Cancer and its diagnostics, Lab on a chip, Drug Discovery		

Suggested Learning Resources: **Textbooks:**	
1	Campbell, N.A.; Reece, J. B.; Urry, Lisa; Cain, M, L.; Wasserman, S.A.; Minorsky, P.V.; Jackson, R. B. Pearson. "Biology: A global approach", , Global Edition, 10/E, 2014
2	David Nelson, Michael Cox. "Lehniger Principles of Biochemistry". W H Freeman & Company, Seventh Edition, 2017.
Reference Books:	
1	Janine M Benvus. "Biomimicry: Innovation inspired by Nature". William Morrow Paperbacks, 2002.
Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning *Real world problem solving and puzzles using group discussion.* *Demonstration of solution to a problem though experiential learning.* Demonstration using real objects taking students on educational tour.	

Course outcomes: **At the completion of the course. The student shall be able to**	
CO1	Apply and utilize essential knowledge of the biological mechanisms of living organisms from the perspective of engineers and find solutions to solve bio-engineering problems with appropriate tools.
CO2	Distinguish and make use of optimal designs in engineering that are bio-mechanical in nature and build and use by observing and understanding bio-physiological processes involved in sensing, locomotion, and knowledge application of range of bio-chemicals.
CO3	Demonstrate that bio-chemical, bio-sensory, bioprocesses could be path-finders to optimise similarities for functional aspects of electronic, computer, mechanical, electrical machines.

CO-PO Mapping												
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**3**	**1**	**1**				**1**					1
CO2	**3**	**1**	**1**				**1**					1
CO3	**3**	**1**	**1**				**1**					1

High-3, Medium-2, Low-1

Course Title	NSS – Phase 1	**Semester**	III
Course Code	NS208	**CIE**	50
Total No. of Contact Hours	25	**SEE**	0
No. of Contact Hours/week	0:2:0	**Total**	50
Credits	0	**Exam. Duration**	-----
Teaching Dept	Any Department		

Course objective is to: National Service Scheme (NSS) will enable the students to:

1.Understand the community in which they work

2.Identify the needs and problems of the community and involve them in problem-solving

3.Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems

4.Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes

5.Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony

Module:1	**RBT Level/s :L1, L2**	**4 Hours**

Youth population in India and its characteristics:

Introduction to India: Physical, Socio-Economic and Demographic Background, Study on Indian Population Composition (Age composition), Youth composition, Youth policy, Importance of Youth Policy, Youth population in India, NSS as a Youth Organization.

Module:2	**RBT Level/s :L1, L2**	**3 Hours**

Fundamentals of NSS:

Introduction to NSS, Origin of NSS, Aims and Objectives of NSS, NSS Motto, NSS Emblem, NSS Badge, NSS Day.

Module:3	**RBT Level/s :L1, L2**	**3 Hours**

NSS Songs:

NSS Anthem (Hindi & Kannada), National Integration song, Rastriya sevayojane Madiharu. Uteh samajkeliye Uteh Uteh. Navellaru Ondagi Balona Banni. Hum Sab Mil ka rDeshka Apani.

Module:4	**RBT Level/s :L1, L2**	**15 Hours**

Activity Based Programmes:

A. Campus Activities:

Shramadhan – Plantation, Cleaning, Watering, Weeding, Any other activities.

Awareness Programmes – Seminar, Workshops, celebration of National and International days, Personality Development Programmes, Group Activities, etc

B : Off Campus Activities:

Rally, Jatha, Visit to Adopted villages, Swatchatha Programme, Visit and Conserving Ancient monuments and heritage site, Socio Economic Survey of village/slum, Nature Camp, Environmental Education

Course outcomes: **At the completion of the course. The student shall be able to**	
CO1	Describe the concept of Youth and compare the international definitions of the term Youth.
CO2	Students will be able to appreciate our demographic advantage and its role in nation building.
CO3	Know the growth and evolution of NSS and its role in Nation building through community service
CO4	Visualize the signs, symbols, logo of NSS and understand their broader meaning.

CO-PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1						1	1					1
CO2						1	1					1
CO3						1	1					1
CO4						1	1					1

High-3, Medium-2, Low-1

Course Title	Physical Education (PE) (Sports and Athletics) – Phase 1	**Semester**	III
Course Code	PE208	**CIE**	50
Total No. of Contact Hours	25	**SEE**	0
No. of Contact Hours/Week	0:2:0	**Total**	50
Credits	0	**Exam. Duration**	------
Teaching Dept	Sports		

Course objective:

Physical education aims to develop all aspects of the human personality through physical and sports activities. Being a subject of science, it has its own value in society and human life. Physical Education is a form of one of the most effective means of education imparted through exercises, fun activities and sports. It is an integral part of the education system. It caters to the need for development of the students on physical, mental and social aspects.

Module – 1	**RBT Level/s :L1, L2**	**5 Hours**

Orientation:
Introduction of Physical Education and sports, Importance of Physical fitness and healthy life style

A. Lifestyle
B. Fitness
C. Food & Nutrition
D. Health & Wellness
E. Pre-Fitness test

Module – 2	**RBT Level/s :L1, L2**	**15 Hours**

General Fitness & Components of Fitness:

A. Warming up (Free Hand exercises)
B. Strength – Push-up / Pull-ups
C. Speed – 30 Mtr Dash
D. Agility – Shuttle Run
E. Flexibility – Sit and Reach
F. Cardiovascular Endurance – Harvard step Test.

Module - 3	**RBT Level/s :L1, L2**	**10 Hours**

Recreational Activities:

A. Postural deformities.
B. Stress management.
C. Aerobics.
D. Traditional Games.

Course outcomes:

At the completion of the course. The student shall be able to

CO1	Understand the fundamental concepts and skills of Physical Education, Health,Nutrition and Fitness

CO2	Familiarization of health-related Exercises, Sports for overall growth anddevelopment
CO3	Create a foundation for the professionals in Physical Education and Sports
CO4	Participate in the competition at regional/state / national / internationallevels..
CO5	Create consciousness among the students on Health, Fitness and Wellness indeveloping and maintaining a healthy lifestyle.

CO-PO Mapping												
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1			1	1					1
CO2			1			1	1					1
CO3			1			1	1					1
CO4			1			1	1					1
CO5			1			1	1					1

High-3, Medium-2, Low-1

Course Title	Yoga for a Better Life - Phase 1	**Semester**	III
Course Code	YG208	**CIE**	50
Total No. of Contact Hours	25	**SEE**	0
No. of Contact Hours/week	0:2:0	**Total**	50
Credits	0	**Exam. Duration**	-----
Teaching Dept			

Course objectives:
1) To enable the student to have good health.
2) To practice mental hygiene.
3) To possess emotional stability.
4) To integrate moral values.
5) To attain higher level of consciousness.

Semester III	**RBT Level/s :L1, L2**	**25 Hours**

Yoga, its origin, history and development. Yoga, its meaning, definitions. Different schools of yoga, Aim and Objectives of yoga, importance of prayer Yogic practices for common man to promote positive health Rules to be followed during yogic practices by practitioner Yoga its misconceptions, Difference between yogic and non yogic practices.

Suryanamaskar prayer and its meaning, Need, importance and benefits of Suryanamaskar12 count, 2 rounds.

Asana, Need, importance of Asana. Different types of asana. Asana its meaning by name, technique, precautionary measures and benefits of each asana Different types of Asanas

a. Sitting
1. Padmasana
2. Vajrasana

b. Standing
1. Vrikshana
2. Trikonasana

c. Prone line
1. Bhujangasana
2. Shalabhasana

d. Supine line
1. Utthitadvipadasana
2. Ardhahalasana

Suggested Learning Resources:
Textbooks:
1. Yogapravesha in Kannada by Ajitkumar
2. Light on Yoga by BKS Iyengar
3. Teaching Methods for Yogic practices by Dr. M L Gharote & Dr. S K Ganguly
4. Yoga Instructor Course hand book published by SVYASA University, Bengaluru
5. Yoga for Children –step by step – by Yamini Muthanna

Web links and Video Lectures (e-Resources):
Refer links
- **https://youtu.be/KB-TYlgd1wE**
- **https://youtu.be/aa-TG0Wg1Ls**

The Health Benefits of Yoga

The benefits of various yoga techniques have been supposed to improve

- body flexibility,
- performance,
- stress reduction,
- attainment of inner peace, and
- self-realization.

The system has been advocated as a complementary treatment to aid the healing of several ailments such as;

- coronary heart disease,
- depression,
- anxiety disorders,
- asthma, and
- extensive rehabilitation for disorders including musculoskeletal problems and traumatic brain injury.

The system has also been suggested as behavioral therapy for smoking cessation and substance abuse (including alcohol abuse). If you practice yoga, you may receive these physical, mental, and spiritual benefits:

- **Physical**
  1. Improved body flexibility and balance
  2. Improved cardiovascular endurance (stronger heart)
  3. Improved digestion
  4. Improved abdominal strength
  5. Enhanced overall muscular strength
  6. Relaxation of muscular strains
  7. Weight control
  8. Increased energy levels
  9. Enhanced immune system
- **Mental**
  1. Relief of stress resulting from the control of emotions
  2. Prevention and relief from stress-related disorders
  3. Intellectual enhancement, leading to improved decision-making skills
- **Spiritual**
  1. Life with meaning, purpose, and direction
  2. Inner peace and tranquility
  3. Contentment

Course outcomes: **At the completion of the course. The student shall be able to**	
CO1	Understand the meaning, aim and objectives of Yoga.
CO2	Perform Suryanamaskar and able to Teach its benefits.
CO3	Understand and teach different types of Pranayama, Asanas by name, its importance, methods and benefits.
CO4	Instruct Kapalabhati and its need and importance.
CO5	Coach different types of Kriyas , method to follow and usefulness.

CO-PO Mapping												
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1			1	1					1
CO2			1			1	1					1
CO3			1			1	1					1
CO4			1			1	1					1
CO5			1			1	1					1

High-3, Medium-2, Low-1

Course Title	Additional Mathematics (Common for all branches)	**Semester**	III
Course Code	DM209	**CIE**	50
Total No. of Contact Hours	30	**SEE**	50
No. of Contact Hours/week	2:0:0	**Total**	100
Credits	0	**Exam. Duration**	3 Hours
Teaching Dept	**MATHEMATICS**		

Course objective is to:

- Develop the knowledge of numerical methods and apply them to solve transcendental and differential equations.
- Study the fundamental concepts of vector calculus viz. Gradient, curl and divergence
- Familiarize the importance of Integral calculus and Linear Algebra.

Module-1: Numerical Analysis	**RBT Levels L1, L2,L3**	**6 Hours**

Solution of algebraic and transcendental equations –Newton-Raphson methods. Finite differences, Interpolation and extrapolation using Newton's forward and backward difference formulae, Newton's divided difference Numerical integration: Trapezoidal rule, Simpson's $(1/3)$rd and $(3/8)$th rules, Weddle's rule.
Numerical Solution of Ordinary Differential Equations: Taylor's series method, Modified Euler's method, Runge-Kutta method of fourth order

Module-2: Ordinary Differential Equations	**RBT Levels L1, L2,L3**	**6 Hours**

Introduction to first-order ordinary differential equations pertaining to the applications for Computer Science & Engineering. Linear and Bernoulli's differential equations. Exact and reducible to exact differential equations - Integrating factors on $\frac{1}{N}\left(\frac{\partial M}{\partial y} - \frac{\partial N}{\partial x}\right)$ $\frac{1}{M}\left(\frac{\partial N}{\partial x} - \frac{\partial M}{\partial y}\right)$.
Higher -order linear ordinary differential equations with constant coefficients - Inverse differential operator,

Module-3: Integral Calculus	**RBT Levels L1, L2,L3**	**6 Hours**

Multiple Integrals: Evaluation of double and triple integrals, evaluation of double integrals by change of order of integration

Module-4: Vector Calculus	**RBT Levels L1, L2,L3**	**6 Hours**

Scalar and vector fields. Gradient, directional derivative, curl and divergence - physical interpretation, solenoidal and irrational vector fields.

Module-5: Linear Algebra	**RBT Levels L1, L2,L3**	**6 Hours**

Row reduction and echelon forms- Consistency of System of Equations. Solution sets of linear equations by Gauss Seidel, Gauss Jordan,. Eigenvalues and eigenvectors Rayleigh's power method.

Suggested Learning Resources: **Textbooks:**	
1.	B.S. Grewal – Higher Engineering Mathematics, Khanna Publishers, 44th edition, 2017.
2.	Erwin Kreyszig - Advanced Engineering Mathematics, Wiley publication, 10th edition, 2015
3	David C Lay: "Linear Algebra and its Applications", Pearson Publishers, 4th Ed., 2018
Reference Books:	
1.	Srimanta Pal & Subodh C Bhunia - Engineering Mathematics, Oxford University Press, 3rd Reprint, 2016.
2.	James Stewart: "Calculus" Cengage Publications, 7thEd., 2019.
3.	Gilbert Strang, Linear Algebra and its Applications, 5th Edition (2016).
4.	N.P Bali and Manish Goyal: "A Textbook of Engineering Mathematics" Laxmi Publications, 10th Ed., 2022. 4. C. Ray Wylie, Louis C. Barrett: "Advanced Engineering Mathematics"
Web links and Video Lectures (e-Resources): **https://youtu.be/Y7VWyyZ6B0g?si=rhxgG4vvrs3VS7mw** **https://youtu.be/zT83sJ5IrEE?si=Crb9_cIWw4tTJxmj** **https://youtu.be/9_m36W3cK74?si=h-bd19yVCgLF3VvW** **https://youtu.be/2DX8Vp1Q2-0?si=rQ76vQyXPATricZz** **https://youtu.be/AuUi_bUeTS4**	
Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning: Course Seminars, Quiz, In class assignments.	

Course outcomes: At the end of the course students will be able to	
CO1	Apply numerical methods to find the solution of algebraic and transcendental equations.
CO2	Apply numerical methods to find the solution of ordinary differential equations.
CO3	Apply the concept of change of order of integration and variables to evaluate multiple integrals and their usage in computing area and volume.
CO4	Use fundamentals of vectors to find gradient, curl and divergence.
CO5	Test the consistency and solve the system of liner equations .

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2		2								2
CO2	3	2		2								2
CO3	3	2		2								2
CO4	3	2		2								2
CO5	3	2		2								2

High-3, Medium-2, Low-1

Course Title	UNIX Shell programming and GIT	**Semester**	III
Course Code	CS209	**CIE**	50
Total No. of Contact Hours	30	**Exam. Duration**	2 Hours
No. of Contact Hours/week	0:2:0:1	**Teaching Dept**	CSE
Credits	0		

Course objective is to:

1. Help the students to understand the effective use of Shell scripting, commands and terminology.
2. Understand UNIX command syntax and semantics.
3. Identify, access, and evaluate UNIX file system.
4. Read and understand specifications, scripts and programs.
5. Familiarize students with basic command of Git
6. Understand how to collaborate and work with Remote Repositories

PREREQUISITES : This course requires that the students are familiar with Commands and programming language such as C programming.

Module-1: Introduction to Shell	**RBT Levels: L1, L2,L3**	**6 Hours**

What is shell scripting? Importance of shell scripting, Different types of shells, creating shell script, making shell scripting executable, Shell input & output, Pipes and Filters, echo, print, read

Module-2: System and File Management	**RBT Levels: L1, L3**	**6 Hours**

Stages of a Process, Types of Process Management, Commands Used to Manage Processes like 'ps','top','kill','nice'.'renice','pkill', 'jobs', 'fg',' bg', Job scheduling commands like 'cron', 'at', 'batch'.
Different types of Files, working with files, working with Directories, basic File Attributes - displaying, searching, comparing, accessing long files and Zip files.

Module-3: User Management	**RBT Levels: L1,L3**	**6 Hours**

Creating Users & Groups, Managing Users and Groups, Create a Group, Modify a Group, Delete a Group, , Create a Account, Modify Account , Delete a Account.

Module-4: Introduction to Git	**RBT Levels:L1,L2,L3**	**6 Hours**

Setting Up and Basic Commands
Initialize a new Git repository in a directory. Create a new file and add it to the staging areaand commit the changes with an appropriate commit message.
Creating and Managing Branches
Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master." Write the commands to stash your changes, switch branches, and then apply the stashed changes.
Collaboration and Remote Repositories

Clone a remote Git repository to your local machine. Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch. Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.

Module-5: Git workflows	**RBT Levels:L3,L4**	**6 Hours**

Git Tags and Releases

Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.

Analysing and Changing Git History

Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message? Write the command to list all commits made by the author "John Doe" between "2023-01-01" and "2023-12-31." Write the command to display the last five commits in the repository's history.

Suggested Learning Resources:

Reference Books:

1.	Unix Shell Programming, Prof: Yogesh HK
2.	Unix Shell Programming, Yashwant Kanetkar
3.	Version Control with Git, 3rd Edition, by Prem Kumar Ponuthorai, Jon Loeliger Released October 2022,
4.	Publisher(s): O'Reilly Media, Inc.
5.	Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, https://git

Web links and Video Lectures (e-Resources):

1. Shell Scripting Tutorial
2. Getting Started with Git

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Create Git Project: Personal Portfolio Website

Objective: Create a Git repository for a personal portfolio website. The repository will include a simple HTML file and showcase basic Git operations such as initialization, staging, committing, branching, and merging.

Course outcomes:

CO1	Ability to understand scripts and programs. Know the basics of Unix concepts and commands.
CO2	Ability to evaluate the UNIX file system and apply Changes in file system.
CO3	Ability to analyze authorizations for different users
CO4	Ability to use the basics commands related to git repository., Create and manage the branches, Apply commands related to Collaboration and Remote Repositories
CO5	Use the commands related to Git Tags, Releases and advanced git operations. Analyse and change the git history.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO2	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO3	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO4	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO5	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3

High-3, Medium-2, Low-1

4th SEMESTER

Course Title	Applied Mathematics-IV for Computer Science and Engineering Stream (AIML, ISE, CSE, CSE-IOT)	**Semester**	IV
Course Code	MC251	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:2:0:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	Mathematics		

Course objective is to understand:

- The concepts of functions and groups
- The basic concepts of graphs and their properties, and operations of graphs
- Hamiltonian and Euler graphs, trees and matrix representation of the graph.

Module-1	**RBT Levels L1,L2,L3**	**8 Hours**

Relations and Functions:

Cartesian Products and Relations, Properties of Relations, Functions – Plain and One-to-One, Onto Functions. Function Composition, and Inverse Functions., Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions.

Module-2	**RBT Levels L1,L2,L3**	**8 Hours**

Group Theory:

Klein 4-group, Additive group of Integers modulo n, Multiplicative group of Integers modulo-p and permutation groups, Properties of groups, Subgroups, cyclic groups, Cosets, Lagrange's Theorem.

Module-3	**RBT Levels L1,L2,L3**	**8 Hours**

Graph Theory :

Directed graphs, Definition –Application of graphs – finite, infinite and bipartite graphs – Incident and Degree – Isolated vertex, Pendant vertex and Null graph.

Paths and circuits- Isomorphism, sub-graphs, walks, paths and circuits, connected graphs, disconnected graphs and components.

Module-4	**RBT Levels L1,L2,L3**	**8 Hours**

Hamilton Cycle and Euler's circuit :

Eulerian and Hamiltonian Graphs-Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Travelling salesman problem.

Digraphs -Types of graphs and binary relation.		
Module-5	**RBT Levels L1,L2,L3**	**8 Hours**
Trees: Rooted Trees, Binary trees, counting Trees and Spanning trees. Pendant vertices in a tree. Connectivity Graphs: Vertex connectivity, Edge connectivity, cut set and cut vertices, Fundamental circuits.		

Suggested Learning Resources: **Textbooks:**	
1	Narsingh Deo, Graph theory with the applications to engineering & Computer Science, Dovers Publications, 2016.
2	J.A. Bondy and U.S.R. Murty. Graph theory with Applications, Springer, 1st edition, 2008.
3	Ralph P. Grimaldi-Discrete and Combinatorial Mathematics, Addision Wesley Publishing company, 2006.
Reference Books:	
1	R. Diestel, Graph Theory, free online edition, 2016: diestel-graph-theory.com/basic.html.
2	Kenneth H. Rosen and Kamala Krithivasn- Discrete Mathematics and Its Applications, Mc graw Hill publication, 2021.
3	Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd.,2010

Web links and Video Lectures (e-Resources):

M1: https://youtu.be/-peAUmrm4RM

M2: https://youtu.be/utBfKsYUwe8

M3: https://youtu.be/tORLeHHtazM

M4: https://youtu.be/V8F8We_-nuo?list=PLhSp9OSVmeyLB62_-fT9VNbjRkDEzJzzp

M5: https://youtu.be/qNqrHO3woyE

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

Quiz, Group Discussions, Seminar, In class assignments

Course outcomes: At the end of the course students will be able to	
CO1	Apply the basic concepts of relations, functions and partially ordered sets for computer representations
CO2	Illustrate the fundamental principles of Algebraic structures with the problems related to computer science & engineering.
CO3	Apply concepts of trees and graph connectivity to solve real world problems.
CO4	Solve the problems involving characterization and operations on graphs.
CO5	Apply concepts of trees and graph connectivity to solve real world problems.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2		2								2
CO2	2	2		2								2
CO3	2	2		2								2
CO4	2	2		2								2
CO5	2	2		2								2

High-3, Medium-2, Low-1

Course Title	Design and Analysis of Algorithms	**Semester**	IV
Course Code	CS252	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:2:0:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

Course objective is to:

1. Enable students to learn the methods for analysing algorithms and evaluating their performance.
2. Enable students to demonstrate the efficiency of algorithms.
3. Prepare students to solve problems using various algorithm design methods
4. Appreciate the concepts of P and NP complexity classes.

PREREQUISITES : This course requires that the students are familiar with programming language and Data Structures and Applications. Graph Theory is desirable

Module-1: Introduction and Overview	**RBT Levels: L1,L2,L3**	**8 Hours**

Introduction and Examples: What is an Algorithm? Algorithm Specification, Examples from real life: Air Travel, Xerox Shop, Document Similarity and types of algorithms.

Motivation for Performance Analysis using Examples: Bubble Sort, Selection Sort, Insertion Sort, String Pattern Matching. Contrast performance analysis versus actual runs.

Performance Analysis Framework: Space complexity, Time complexity. **Asymptotic Notations:** Big-Oh notation (*O*), Omega notation (Ω), Theta not - ation (□), Mathematical analysis of non recursive and recursive Algorithms with Examples**.**

Text Book 1: Chapter 1.1,2.1-2.4,3.1,3.2 Digital Resource: D1

Module-2: Decrease and Conquer, Divide and Conquer, Greedy Method	**RBT Levels: L1,L2,L3**	**8 Hours**

Decrease and Conquer Approach: Insertion sort**,** Topological Sort**.**

Divide and Conquer: General method, Recurrence equation for divide and conquer, Finding the maximum and minimum. Merge sort, Quick sort

Greedy Method: General method, Knapsack Problem, Job sequencing with deadlines,

Text book 1: Chapter 4.1,4.2,5.1,5.2,6.4 Text book 2:4.1,4.3,4.5

Module-3: Greedy Method, Transform and Conquer, Dynamic Programming	**RBT Levels:L1,L2,L3**	**8 Hours**

Minimum cost spanning trees: Prim‟s Algorithm, Kruskal‟s Algorithm, union find method

Single source shortest paths: Dijkstra's Algorithm. **Optimal Tree problem:** Huffman Trees and Codes. **Transform and Conquer Approach:** Heaps and Heap Sort.

Dynamic Programming: General method with Examples, **Transitive Closure:** Warshall‟s Algorithm, **All Pairs Shortest Paths:** Floyd's Algorithm

Text book 1: 9.1,9.2,9.3,9.4,8.1,8.4		
Module-4: Dynamic Programming, Backtracking	**RBT Levels:L1,L2,L3**	**8 Hours**
Dynamic Programming (cont…): Multistage Graphs, Travelling Sales Person problem, Knapsack problem, Bellman-Ford Algorithm, **Backtracking:** General method, N-Queens problem, Sum of subsets problem, Hamiltonian cycles. **Text book 1: 8.2,12.1 text book 2: 5.2, 5.4, 5.9**		
Module-5: Space and Time Trade-offs, Branch and Bound and Case Studies	**RBT Levels:L1,L2,L3**	**8 Hours**
Counting Sort Branch and Bound: Assignment Problem, Travelling Salesperson problem, 0/1 Knapsack **NP-Complete and NP-Hard Problems concept**. Case Studies: Efficient Route calculation application in GPS navigation system: **The navigation system calculates the optimal route from the user's current location to the destination. This calculation considers factors such as distance, estimated travel time, traffic conditions.** **Summarization of all modules.** **Text book 1: 7.1, 12.2, 12.3**		
Follow on Courses: Advanced Algorithms, Machine Learning		
Resources for Advanced Learners: 1. Data Structures and Algorithms Made Easy – Data Structures & Algorithmic Puzzles Author: Narasimha Karumanchi (M.Tech IIT Bombay, Founder- CareerMonk.com) 5 th Edition 2. Algorithms - Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani published by Mc Graw Hill		

Suggested Learning Resources: **Textbooks:**	
1.	Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009. Pearson.
2.	Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press
Reference Books:	
1.	Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
2.	Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education).
Web links and Video Lectures (e-Resources): 1. Design and Analysis of Algorithms course by Madhavan Mukund	

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

1. Real world problem solving and puzzles using group discussion. E.g., Fake coin identification,Peasant, wolf, goat, cabbage puzzle, Konigsberg bridge puzzle etc.,

2. Demonstration of solution to a problem through programming.

Course outcomes:	
CO1	Apply asymptotic notational method to analyse the performance of the algorithms in terms of time complexity
CO2	Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems
CO3	Use transform & conquer and dynamic programming methodologies to solve the given real world computational problems
CO4	Explain various classes (P, NP and NP Complete) of problems and demonstrate backtracking, branch & bound and approximation methods.
CO5	Apply appropriate algorithm design strategies to a given case study/use case.

CO-PO Mapping															
CO/ PO	PO1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO2	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO3	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO4	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-
CO5	3	3	3	3	-	-	-	-	1	1	-	-	3	-	-

High-3, Medium-2, Low-1

Design & Analysis of Algorithms Laboratory			
Course Code	**CS253**	**Semester**	**IV**
L:T:P:S	**0:0:2:1**	**CIE Marks**	**50**
Hrs. / Week	**2**	**SEE Marks**	**50**
Credits	**1**	**Total Marks**	**100**
Exam Hours	**03**		

Course objectives : This laboratory course enables students to	
1.	Train students to design and implement various algorithms in JAVA/C/C++
2.	Assist students to employ various design strategies for problem solving.
3.	Enable students to measure and compare the performance of different algorithms.

Pgm. No.	**List of Experiments / Programs**	**Hours**	**COs**
Prerequisite Experiments / Programs / Demo			
This course requires that the students are familiar with programming language and Data Structures & Applications. Graph Theory is desirable.			
PART-A			
1	**Title:** Program that uses simple technique to a) search a key b) to sort n elements **Problem Description:** Run the program for varied values of n>5000 and record the time taken to search/sort. Plot a graph of the time taken v/s n. The elements can be read from a file or can be generated using the random number generator. **Method:** A Brute Force- Non recursive algorithm implementation. Sequential Search and Selection Sort. **Theory Reference:** Module 1	2	CO1,CO3
2	**Title**: Sorting elements based on their value. **Problem Description:** Run the program for varied values of n to demonstrate the behaviour of the algorithm in the Worst, Bestand Average Cases. Record the time taken to sort. Plot a graph of the time taken versus *n* on graph sheet. The elements can be read from a file or can be generated using the random number generator for large values of n. **Method:** A Divide & Conquer approach partitioning the dataset on a pivot- Recursive algorithm implementation. **Theory Reference:** Module 2	2	CO1,CO3

3	**Title:** Sorting elements in a list by breaking them into sub-lists. **Problem Description:** Run the program for varied values of n to demonstrate the behaviour of the algorithm in the Worst, Bestand Average Cases. Record the time taken to sort. Plot a graph of the time taken versus *n* on graph sheet. The elements can be read from a file or can be generated using the random number generator for large values of n. **Method:** A Divide & Conquer- Recursive algorithm implementation- Merge Sort **Theory Reference:** Module 2	2	CO1,CO3
4	**Title:** Urban planning of water supply networks **Problem Description:** Design an optimized water distribution network that minimizes the total cost of the infrastructure while ensuring reliable water supply to all areas in the given region. **Method:** Represent the problem in the form of a graph. Use the greedy Technique for a spanning tree which has a total minimum weight. **Theory Reference:** Module 3	2	CO2,CO3
5	**Title:** Optimal Road Network **Problem Description:** A local government wants to improve the road network between towns to enhance connectivity and reduce travel costs. The region consists of five towns that need to be connected by the most efficient road network possible, minimizing the total length of the roads while ensuring that each town is accessible from any other town. **Method:** A greedy Technique for a spanning tree which has atotal minimum weight. Use Union-Find Data Structure to detect cycles during the execution of the algorithm. **Theory Reference:** Module 3	2	CO2,CO3

	PART-B		
6	**Title:** Optimal Road Routes **Problem Description:** You are the chief planner for a transportation department in a country with several cities. The cities are connected by a network of roads, each with a specific travel time (in hours). Your task is to determine the shortest travel time from a designated source city to all other cities. **Method:** Represent the problem in the form of a graph using adjacency matrix or adjacency list. Using the Single Source Shortest Path Algorithm find the shortest path from the source city to all other cities. **Theory Reference:** Module 3	2	CO2,CO3
7	**Title:** Metropolitan Subway System Optimization **Problem Description:** A metropolitan city is planning to optimize its subway system to ensure efficient travel for its commuters. The subway system consists of various stations connected by subway lines, and each line has a different travel time. The city's transportation authority wants to find the shortest travel time between all pairs of stations. **Method:** Use Dynamic programming technique to find the shortest travel time between all pairs of stations using the Floyd-Warshall algorithm **Theory Reference:** Module 3	2	CO2,CO3
8	**Title:** Optimal Product Selection for a Limited Shelf Space **Problem Description:** You are a store manager responsible for stocking products on the shelves. The store has limited shelf space, and you want to maximize the total profit by selecting the most valuable products to display. Each product has a specific weight and profit associated with it. **Method:** Solve the 0/1 Knapsack Problem using dynamic programming or other optimization techniques. The key idea is to build a table that stores the maximum value achievable for different weights and items. **Theory Reference:** Module 4	2	CO2,CO3

9	**Title:** Event Budget Allocation **Problem Description:** An event planner is organizing a conference and has a fixed budget to allocate across various categories such as venue, catering, speakers, and marketing. Theplanner needs to ensure that the total expenditure does notexceed the budget while maximizing the quality of the event. Setof Expenses (with costs): Venue rental-$5000, Catering-$2000, Keynote Speaker-$3000, Marketing-$1500, Audio/Visual Equipment-$1000. Given Fixed budget-$8000 **Method:** Use backtracking technique to solve the problem. **Theory Reference:** Module 4	2	CO2,CO3
10	**Title:** Delivery Route Optimization. **Problem Description:** Imagine a delivery company that operates in a city with five key delivery locations. The company needs to design a route for their delivery vehicle that visits each location exactly once and returns to the starting point without retracing its steps. **Method:** Given a graph representing the locations as vertices and the roads between them as edges, determine whether a Hamiltonian Circuit exists. If it does, find the circuit. **Theory Reference: Module 4**	2	CO2,CO3

PART-C

Beyond Syllabus Virtual Lab Content

1. N-Queens problem
2. Topological Sorting
3. Heap Sort

Suggested Learning Resources:

Textbooks:

1	Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, 2009.Pearson.
2	Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press

Web links and Video Lectures (e-Resources):

1. http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS43.html
2. https://nptel.ac.in/courses/106/101/106101060/
3. http://elearning.vtu.ac.in/econtent/courses/video/FEP/ADA.html
4. http://cse01-iiith.vlabs.ac.in/
5. http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

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

Course outcomes:	
CO1	Choose appropriate algorithm design techniques like divide and conquer, Decrease and conquer, Transform and conquer to develop solutions to the computational and complex problems.
CO2	Develop programs to solve computational problems using greedy method, Dynamic programming, Backtracking algorithm design strategies.
CO3	Compare algorithm design strategies by developing equivalent programs and observing running times for empirical analysis.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	3	3	-	-	-	-	-	-	1	3	-	-
CO2	3	3	3	3	3	-	-	-	-	-	-	1	3	-	-
CO3	3	3	3	3	3	-	-	-	-	-	-	1	3	-	-

Course Title	Microprocessor & Microcontrollers	**Semester**	IV
Course Code	CS254	**CIE**	50
Total No. of Contact Hours	50	**SEE**	50
No. of Contact Hours/week	3: 0: 2:2	**Total**	100
Credits	4	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

PREREQUISITES: **Digital Design and Computer Organization, C Programming.**

Course objective is to:

- Enable students to understand the fundamentals of 8086 Microprocessors.
- Assist students to familiarize themselves with 8086 Microprocessor programming.
- Help students to understand the fundamentals of ARM Microcontroller.
- Train students to develop Assembly Level Programming using various instruction sets to program the Microcontrollers.
- Prepare students to learn New Architecture and engage in a Case study.

Module-1: Overview and Architecture of 8086	**RBT Levels:L1, L2**	**10 Hours**

Microprocessor vs Microcontroller, Applications of Microcontrollers, contemporary study on different microprocessors and microcontrollers (8051, PIC, ARM),
Introduction of 8086: Register Organization of 8086,8086 microprocessor architecture, Machine language Instruction format, Addressing modes of 8086.

Text Book 1: 1.1,1.2,2.1,2.2

Module-2: Instruction Set of 8086	**RBT Levels:L1, L2, L3**	**10 Hours**

Instruction set of 8086, Assembler directives and operators, A few Machine Level Program, Machine Coding the Program, Assembly language example program

Text Book 1: 2.3,2.4,3.1,3.2,3.4

Module-3: Introduction to ARM	**RBT Levels:L1, L2, L3**	**10 Hours**

The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software. Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table, Core Extensions

Textbook 2: Chapter 1 - 1.1 to 1.4, Chapter 2 - 2.1 to 2.5

Module-4: ARM Instruction Set	**RBT Levels:L1, L2, L3**	**10 Hours**

Data Processing Instructions, Branch Instructions, Load-Store Instructions, Software Interrupt Instructions, Program Status Register Instructions, Loading Constants.

Textbook 2: Chapter 3 - 3.1 to 3.6

Module-5: Emerging Architecture: RISC-V	RBT Levels:L1, L2	10 Hours

Introduction, Assembly Language, Programming, Machine Language (R-type, I-type, S/B type, U/J-type instructions), Evolution of the RISC-V.

Case Study: Ibex core for real time applications.

Textbook 3: Chapter 6 – 6.1 to 6.4 and 6.7

Follow-on Courses: Embedded Systems, Internet of Things, Advanced Computer Architecture.

Topics for Advanced Learners:

- Advanced RISC-V
- Embedded System Design

Resources for Advanced Learners:

David A. Patterson & John L. Hennessy, Computer Organization and Design RISC-V Edition.
https://www.youtube.com/watch?v=GySwSvU3Ak4
https://elearn.nptel.ac.in/shop/nptel/introduction-to-embedded-system-design/?v=c86ee0d9d7ed

Suggested Learning Resources:

Textbooks:

1.	Advanced Microprocessors and Peripherals, K M Bhurchandi, A K Ray, 3rd Edition Carl Hamacher, Tata McGraw Hill.
2.	Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, Morgan Kaufman publishers, 2008.
3.	Sarah L. Harris, David Harris, Digital Design and Computer Architecture RISC-V Edition.
Reference Books:	
1.	The Intel Microprocessors, Barry B. Brey,8th Edition, Prentice Hall
2.	The Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005.
3.	David A. Patterson & John L. Hennessy, Computer Organization and Design RISC-V Edition.

Web links and Video Lectures (e-Resources):
https://archive.nptel.ac.in/courses/106/105/106105193/
https://riscv.org/
https://onlinecourses.nptel.ac.in/noc23_cs113/preview

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

4. Flipped class room
5. Presentation

Course outcomes:	
CO1	To understand 8086 Microprocessor Architectural features.
CO2	To use 8086 Microprocessor Instruction set.

CO3	To understand ARM Architectural features.
CO4	To use ARM Instruction Set.
CO5	To understand RISC-V Architecture and apply Instruction Set.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	3	-	2	2	-	-	-	-	-	2	3	3	-
CO2	2	3	3	-	2	2	-	-	-	-	-	2	3	3	-
CO3	2	3	3	-	2	2	-	-	-	-	-	2	3	3	-
CO4	1	3	3	-	2	2	-	-	-	-	-	2	3	3	-
CO5	1	3	3	-	2	2	-	-	-	-	-	2	3	3	-

High-3, Medium-2, Low-1

Microprocessor and Microcontroller Laboratory			
Course Code	CS255	**Semester**	IV
L:T:P:S	0:0:2:1	**CIE Marks**	50
Hrs. / Week	2	**SEE Marks**	50
Credits	1	**Total Marks**	100
Exam Hours	03		

Course objectives: This laboratory course enables	
1.	To introduce students with the architecture and instruction set of x86 microprocessors and ARM microcontrollers.
2.	To familiarize the students with the programming of x86 microprocessors and ARM microcontrollers.
3.	To learn interfacing of microcontrollers.

Pgm. No.	**List of Experiments / Programs**	**Hours/Week**	**COs**
	PART-A		
	Write Assembly Language program using x86 instructions for the following and execute the same.		
1	Read two strings, store them in locations STR1 and STR2. Check whether they are equal or not and display appropriated messages. Also display the length of the stored strings.	2	CO1
2	Simulate a Decimal Up-counter to display 00-99.	2	CO1
3	Compute **nCr** using recursive procedure. Assume that ‘n’ and ‘r’ are non- negative integers	2	CO1
4	Sort a given set of ‘n’ numbers in ascending and descending orders using the Bubble Sort algorithm.	2	CO1
5	Read the current time from the system and display it in the standard format on the screen.	2	CO1
	PART-B		
	Conduct the following experiments on an ARM7TDMI/LPC2148 evaluation board using evaluation version of Embedded 'C' & Keil Uvision-4 tool/compiler.		
6	i)Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction. ii)Interface and Control a DC Motor.	2	CO2,CO3
7	Display “Hello World” message using Internal UART.	2	CO2, CO3

8	Display the 4 digit counter sequence 000, 001, ….FFF on a 7-segment LED interface, with an appropriate delay in between .	2	CO2, CO3
9	Determine Digital output for a given Analog input using Internal ADC of ARM controller.	2	CO2, C 3
10	Interface a 4x4 keyboard and display the key code on an LCD.	2	CO2,CO3

Conduct of Practical Examination:

Experiment distribution : Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

Mark distribution: Part A = 40% Marks,Part B =60 % Marks

Course outcomes:

CO1: Understand and apply the fundamentals of assembly level programming of x86 microprocessors.

CO2: Develop embedded C programs for ARM microcontrollers.

CO3: Understand the interfacing of ARM microcontrollers in real world applications.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	2	2	--	-	-	2	-	2	-	-	2	2	2
CO2	1	2	2	2	-	-	-	2	-	2	-	-	1	1	1
CO3	1	2	2	2	-	-	-	2	-	2	--		2	2	2

Course Title	Web Programming	**Semester**	IV
Course Code	CS256	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	2:0:2:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

Course objective is to:

1. Enable the students to design layouts of web pages.
2. Enable students to Client and Server-Side Programming
3. Enable students to understand Data-base Connectivity aspects in web applications.
4. Enable students to add dynamic content to web pages.
5. Enable students to develop end-to-end web applications.

PREREQUISITES : Basic Programming Skills

Module-1: Introduction and Overview	**RBT Levels:L1,L2,L3**	**8 Hours**

Web Basics and Overview: Introduction to Internet, World Wide Web, Web Browsers, URL, MIME, HTTP, Web Programmers Toolbox.

Introduction to HTML: What is HTML and Where did it come from? ,HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, frames

HTML5 Semantic Structure Elements: Header and Footer, Heading Groups, Navigation, Articles and Sections, figure and Figure captions, Aside.

Textbook: T1

Experiential Learning: (Experiments which can be conducted on the concepts of contents)

1. Write an HTML program for the demonstration of Lists. a. Unordered List b. Ordered List c. Definition List d. Nested List.
2. Write an HTML program for demonstrating Hyperlinks. a. Navigation from one page to another. b. Navigation within the page.
3. Write an HTML program for timetable generation using tables.
4. Write an HTML program to develop a static Home Page using frames.
5. Write an HTML program to develop a static Registration Form.
6. Write an HTML program to develop a static Web Page for Shopping Cart.

HTML5:
https://www.studocu.com/row/document/national-textile-university/web-development/lecture-4-5-html-5-lab-manual/48995731

Module-2: Layout Design of Web Pages	RBT Levels:L1,L2,L3	8 Hours

Cascading Style Sheets: Introduction to CSS. What is CSS? CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling.

Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.

Textbook: T1

Experiential Learning: (Experiments which can be conducted on the concepts of contents)

1. Write HTML for demonstration of cascading stylesheets. a. Embedded stylesheets. b. External style sheets. c. Inline styles.

2. Design a web page using CSS which includes the following: I. Use different font and text styles ii. Set a background image for both the page and single element on the page. iii. Define styles for links iv. Working with layers v. Adding a Customized cursor

Module-3: Client-Side Scripting	RBT Levels:L1,L2,L3	8 Hours

JavaScript: client-side scripting

What Is JavaScript and What can it do? JavaScript Design principles, where does script go? Syntax, JavaScript object, form

Textbook:T1

Experiential Learning: (Experiments which can be conducted on the concepts of contents)

1. Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference, and quotient.
2. Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs HTML text that displays the resulting values in an HTML table format.
3. Write a JavaScript code that displays text "TEXT-GROWING" with increasing font size in the interval of 100 ms in RED COLOR, when the font size reaches 50 pt it displays "TEXT-SHRINKING" in BLUE colour. Then the font size decreases to 5 pt.
4. Develop and demonstrate a HTML5 file that includes JavaScript script that uses functions for the following problems:
    1. Parameter: A string
    2. Output: The position in the string of the left-most vowel
    3. Parameter: A number
    4. Output: The number with its digits in the reverse order

5. Write a JavaScript program to validate USER LOGIN page.

6. Write a JavaScript program for validating REGISTRATION FORM

Module-4: Server-Side Development	RBT Levels:L1,L2,L3	8 Hours

Introduction to Server-Side Development with PHP: What Is Server-side Development? Quick Tour of PHP, Program control, function, PHP arrays and Super-global: Arrays, S_GET and $_POST super global Arrays.

Textbook T1

Develop and demonstrate PHP Script for the following problems:

1. Write a PHP Script to find out the Sum of the Individual Digits.
2. Write a PHP Script to check whether the given number is Palindrome or not

1. Write a program using PHP and HTML to create a form and display the details entered by the user

2. Write a PHP program to store current date-time in a COOKIE and display the 'Last visited on' date-time on the web page upon reopening of the same page.

3. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

Module-5: Web Application Development	**RBT Levels:L2,L3**	**8 Hours**

Working with Databases:

Databases and Web Development, SQL, PHP. MySQL API, Accessing MySQL in PHP.

Textbook: T1

Experiential Learning: (Experiments which can be conducted on the concepts of contents)

1. Create an HTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.

2. Write a PHP program to insert the details entered by the user in the Registration form into MySQL database.
3. Activity Based Learning (Suggested Activities in Class)/ Practical Based learning Mini Project:

Case Study:
Photo Sharing Web Application among friends and relatives (after travel, special occasions)

Follow on Courses: Mobile Application and Web Technologies, Frameworks and MVC (AngularJS), React, python (Django, Flask, pip)

Resources for Advanced Learners:

1. PROGRAMMING THE WORLD WIDE WEB – Robert W Sebesta, 4th Edition, Pearson Education, 2008
2. David Sawyer Mcfarland, "JavaScript & jQuery: The Missing Manual", 1 st Edition, O"Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014
3. Uttam K.Roy, "Web Technologies", Oxford publishers.
4. An Introduction to Web Design and Programming –Wang-Thomson
5. JavaScriptTM for programmers Deitel Developer series , Indian edition published by Dorling Kindersley India Pvt Ltd, Copyright 2010.

Suggested Learning Resources:

Textbooks:

1.	Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1st Edition, Pearson Education India. (ISBN:978-9332575271)

Web links and Video Lectures (e-Resources):

1. Web Development (w3schools.com)
2. Web Technology - GeeksforGeeks
3. Web Technology - Course (swayam2.ac.in)

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

Mini Project

Course outcomes:	
CO1	Adapt HTML and CSS syntax and semantics to build web pages.
CO2	Construct and visually format tables and forms using HTML and CSS
CO3	Develop Client-Side Scripts using JavaScript
CO4	Develop Server-Side Scripts using PHP to generate and display the contents dynamically.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	2	3	2	2
CO2	3	2	3	-	-	-	-	-	-	-	-	2	3	2	2
CO3	3	2	3	-	2	-	-	-	-	-	-	2	3	2	2
CO4	3	2	3	1	2	-	-	-	1	1	-	2	3	2	2

High-3, Medium-2, Low-1

Course Title	Finite Automata and Formal Language	**Semester**	IV
Course Code	CS261	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	3:0:0:1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept	CSE/ISE		

Course objective is to:

1. To introduce the concepts of formal languages, grammar, and finite automata.
2. Enable students to learn and apply formal mathematical models to prove Grammar and Automata.
3. Help students to understand and apply finite state machines to solve problems in computing.
4. Make students understand the limitations of computation.

PREREQUISITES : This course requires that the students are familiar with Discrete Mathematical Structures, Graph Theory, and Analysis & Design of Algorithms (BCS401)

Module-1: Overview of Automata Theory	**RBT Levels:L1,L2**	**7 Hours**

Mathematical Preliminaries: Sets, Graphs & Trees, Proof Techniques.

Motivation: Modelling Hardware and Software Systems as Finite State Machines. A high-level introduction to Automata theory and its applicability to various domains.

Introduction to Automata: The Central Concepts of Automata Theory, Chomsky Hierarchy.

Digital Resource: D1. Textbook: 2 (1.1, 1.2, 11.4)

Module-2: Introduction to Finite Automata and Regular Languages	**RBT Levels:L1,L2**	**9 Hours**

Finite Automata: Deterministic Finite Automata; Nondeterministic Finite Automata; Conversion of NFA to DFA; Finite Automata with Epsilon-Transitions: Epsilon Closure, Conversion of Epsilon NFA to DFA, Equivalence and Minimization of Automata: Minimization using Table Filling Method.

Regular Expression and Languages: Regular Expressions; Finite Automata and Regular Expressions: Conversion using State Elimination & Kleene's Theorem, Conversion of Regular Expression to DFA.

Textbook:1 (2.1, 2.2, 2.3, 2.4, 3.1, 3.2**)**

Module-3: Properties of Regular Languages and Context-Free Grammars	**RBT Levels:L1,L2,L3**	**9 Hours**

Properties of Regular Languages: Proving Languages not to be Regular Languages using Pumping Lemma.

Context-Free Grammars and Languages: Context-Free Grammars. Parse trees, Ambiguity in Grammars and Languages, Simplification of Context-free Grammars

Textbook: 1 (4.3, 5.1, 5.2, 6.1)

Module-4: Properties of Context-Free Languages & Pushdown Automata	RBT Levels:L1,L2,L3	7 Hours

Properties of Context-Free Languages: Normal forms for CFGs (CNF & GNF).

Pushdown Automata: Definition of the Pushdown Automata; Deterministic Pushdown Automata.

Textbook: 1 (6.2, 7.1, 7.2)

Module-5: Turing Machine, Decidability and Complexity	RBT Levels:L1,L2,L3	8 Hours

Introduction To Turing Machine: The Turning Machine, Turing Thesis, Variants of Turing Machine, Turing Machine with Complex Storage, Halting Problem

Decidability: Recursive Language & Recursively Enumerable Language; Post's Correspondence problem.

Complexity: Efficiency of Computation, The Classes P and NP, Some NP Problems.

Any one of the following Case Studies:

Case Study: Elevator Control System

Model an elevator control system for a multi-floor building using automata theory. The system must handle requests from different floors efficiently, incorporating static and dynamic policies for halting. Also, it should dynamically adjust its behavior based on the load inside the elevator. Additionally, the system must handle emergencies such as power outages or system failures, ensuring safety and reliability. The automata involved should include components for request handling, movement control, door management, floor indication, load management, and emergency handling. Analysis can consider inflow and outflow on various floors and suggest better policies for lift operation.

Case Study: Network Control

Consider a scenario where a network experiences link failures due to hardware faults or network congestion. Model a Finite automata that can adapt to these changes by rerouting traffic along alternate paths, avoiding failed links and minimizing packet loss and latency. The finite automata representing routers continuously update their states and routing tables based on feedback from neighboring routers and network monitoring systems, ensuring that data packets reach their destinations efficiently despite the network disturbances.

Case Study: Natural Language Processing

Consider problem of tokenization with FSA for NLP that breaks down raw text into manageable units (words, numbers, punctuation) for further processing and analysis. The goal is to model a single, efficient FSA for text tokenization, focusing on efficient character separation and improvement at each step. Build an FSA to recognize individual characters, modify it to differentiate words from non-word characters based on transitions, and then modify it to identify and separate number sequences.

Textbook:1 (9.1, 9.3, 10.1, 10.2, 10.4, 10.5, 11.1, 12.1, 12.3, 14.1, 14.4, 14.5)**, Reference book: 1**

Follow on Courses: Compiler Design, Natural Language Processing

Resources for Advanced Learners:

1. Kavi Mahesh, Theory of Computation: A Problem-solving Approach, John Wiley & Sons, Limited, 2012.

Suggested Learning Resources:

Textbooks:

1.	Peter Linz, An introduction to formal languages and automata, Jones and Bartlett Learning, 4th Edition, 2009.
2.	Christel Baier, Joost-Pieter Katoen, Principles of Model Checking, The MIT Press Cambridge, Massachusetts London, England, 2008

Reference Books:

1.	Elain Rich, "Automata, Computability and Complexity: Theory and Applications", Pearson Prentice Hall, 2019.

Web links and Video Lectures (e-Resources):

NPTEL course on Model Checking by Prof. Srivathsan B.

https://onlinecourses.nptel.ac.in/noc20_cs38/preview

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

Flipped Class room

Course outcomes:	
CO1	Understand the fundamental concepts of computational theory.
CO2	Analyze the usage of finite automata in different areas of computer science.
CO3	Design the models for various complex problems.
CO4	Apply formal mathematical models to prove properties of formal languages

CO-PO Mapping

CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO 3
CO1	2	2	3	2	-	3	1	1	2	1	1	1	2	2	2
CO2	2	2	3	2	-	3	1	1	2	1	1	1	2	2	2
CO3	2	2	3	2	-	3	1	1	2	1	1	1	2	2	2
CO4	2	2	3	2	-	3	1	1	2	1	1	1	2	2	2

High-3, Medium-2, Low-1

Course Title	Optimization Techniques	**Semester**	IV
Course Code	CSE262	**CIE**	50
Total No. of Contact Hours	40	**SEE**	50
No. of Contact Hours/week	3:0:0.1.5	**Total**	100
Credits	3	**Exam. Duration**	3 Hours
Teaching Dept.	CSE/ISE		

Prerequisites: Engineering Mathematics

Course objectives:

1. The objective of this course is to make students to learn principles of optimization
2. To implement the optimization Concepts for the structural engineering problems.
3. To evaluate different methods of optimization.

Modules

Module-1	**RBT Level: L1, L2**	**8 Hours**

Introduction: Introduction to optimization, engineering applications of optimization, Formulation of structural optimization problems as programming problems. Optimization Techniques: Classical optimization techniques, single variable optimization, multivariable optimization with no constraints, unconstrained minimization techniques and algorithms constrained optimization solutions by penalty function techniques, Lagrange multipliers techniques and feasibility techniques.

Module-2	**RBT Level: L1, L2, L3**	**8 Hours**

Linear Programming: Linear programming, standard form of linear programming, geometry of linear programming problems, solution of a system of linear simultaneous equations, pivotal production of general systems of equations, simplex algorithms, revised simpler methods, duality in linear programming.

Module-3	**RBT Level: L2, L3**	**8 Hours**

Non-linear programming: Non-linear programming, one dimensional minimization methods, elimination methods, Fibonacci method, golden section method, interpolation methods, quadratic and cubic methods, Unconstrained optimization methods, direct search methods, random search methods, descent methods

Module-4	**RBT Level: L2, L3. L4**	**8 Hours**

Constrained optimization techniques such as direct methods, the complex methods, cutting plane method, exterior penalty function methods for structural engineering problems. Formulation and solution of structural optimization problems by different technique

Module-5	**RBT Level: L2, L3. L4**	**8 Hours**

Geometric programming: Geometric programming, conversion of NLP as a sequence of LP/ geometric programming. Dynamic programming: Dynamic programming conversion of NLP as a sequence of LP/ Dynamic programming

Course outcomes:

On completion of this course, students are able to:

CO1:Achieve Knowledge of design and development of problem solving skills.

CO2: Understand the principles of optimization.

CO3: Design and develop analytical skills.

CO4: Summarize the Linear, Non-linear and Geometric Programming

CO5: Understands the concept of Dynamic programming

Question paper pattern:

The question paper will have ten questions.

There will be 2 full questions (with a maximum of four sub questions) from eachmodule.

Each full question will have sub questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from eachmodule.

Reference Books:

1. Spunt, "Optimum Structural Design"- Prentice Hall
2. S.S. Rao, "Optimization – Theory and Practice"- Wiley Eastern Ltd.
3. Uri Krisch, "Optimum Structural Design"- McGraw Hill
4. Richard Bronson, "Operation Research"- Schaum's Outline Series
5. Bhavikatti S.S.- "Structural optimization using sequential linear programming"- Vikas publishing house

Web links and Video Lectures (e-Resources):

https://www.youtube.com/watch?v=wEdZLKMMZ8o&list=PLwdnzlV3ogoXKKb9nABDWYltTDgi37lYD
https://www.youtube.com/watch?v=GMTvoKRfxQw&list=PLGbjwqYC00hsy6XGalOBAphm2tdeLbgK0
https://www.youtube.com/watch?v=fszNBvdfKrY

Skill Development Activities Suggested

- **Conduction of technical seminars on recent research activities**
- **Group Discussion**

	PO										
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	3		3					3	3	3
CO2	3	3		3					3	3	3
CO3	3	3		3					3	3	3
CO4	3	3		3					3	3	3
CO5	3	3		3					3	3	3

Course Title	Special Topic	**Semester**	IV
Course Code	CS271	**CIE**	50
Total No. of Contact Hours	30	**SEE**	50
No. of Contact Hours/Week	2:0:0:1	**Total**	100
Credits	2	**Exam. Duration**	3 Hours
Teaching Dept.	CSE/ISE		

Course objective: To enable students

1. Explore and understand emerging technologies and research domains.
2. Develop skills in selecting and defining a special topic within emerging technologies.
3. Conduct in-depth study and analysis of a selected special topic.
4. Design and develop a proof-of-concept, prototype, or study paper related to the special topic.

Module 1: Exploration into Emerging Technologies /Research Domains	**RBT Levels: L1, L2**	**6 Hours**
Module 2: Selection of Special Topic	**RBT Levels: L1, L2**	**6 Hours**
Module 3: Exploration & Study of Special Topic	**RBT Levels: L1, L2**	**6 Hours**
Module 4: Preparation of Proof-of-Concept, Prototype or Study Paper Related to Special Topic	**RBT Levels: L1 ,L2,L3**	**6 Hours**
Module 5: Demonstration & Report	**RBT Levels: L1, L2**	**6 Hours**

Suggested Learning Resources:
1. Online resources and articles
2. Guest lectures from industry experts

Course outcomes:
At the completion of the course. The student shall be able to

CO1	Identify and explain current and future trends in emerging technologies
CO2	Conduct thorough research and analysis on a selected special topic.
CO3	Apply critical thinking, problem-solving, and communication skills in the context of emerging technologies
CO4	Design and develop innovative solutions, prototypes, or study papers related to the special topic

Indicative Topics

Domains	**Sample Topics**
Computer Vision & Image Processing	Deep Learning for Computer Vision, Image Understanding and Analysis, Computer Vision for Robotics and Autonomous Systems, Medical Image Analysis, 3D Computer Vision, Edge AI for Computer Vision.
Soft Computing	Fuzzy Logic and Systems, Neural Networks and Deep Learning, Soft Computing for Image Processing, Soft Computing for Natural Language Processing, Hybrid Soft Computing Models, Soft Computing for Robotics and Autonomous Systems.

Cloud & Edge Computing	Cloud Infrastructure and Architecture, Cloud Security and Compliance, Cloud Migration and Deployment, Edge AI and Machine Learning, Edge Computing for IoT and 5G, Cloud-Edge Application Development.
Cyber Security & TC	Cyber Security and TC for Smart Cities, Cyber Security and TC for Autonomous Vehicles, Cyber Security and TC for Autonomous Vehicles.
Software Architecture & Information Management	Digital Twin Architecture, Serverless Architecture and Computing, Blockchain Architecture and Development, Software Architecture for Edge Computing, Information Management for IoT and Edge Computing
Programming Paradigm & its Applications	Object-Oriented Programming, Functional Programming, Declarative Programming, Event-Driven Programming

Note: A dedicated teacher facilitator will be allotted to coordinate and to track progress, while students will be mentored by expert guides from their chosen domains, ensuring a comprehensive and industry-informed learning experience.

Course Title	UNIX Shell programming and GIT	**Semester**	IV
Course Code	CS272	**CIE**	50
Total No. of Contact Hours	30	**Exam. Duration**	2 Hours
No. of Contact Hours/week	2:2:0:1	**Teaching Dept**	CSE/ISE
Credits	2		

Course objective is to:

7. Help the students to understand the effective use of Shell scripting, commands and terminology.
8. Understand UNIX command syntax and semantics.
9. Identify, access, and evaluate UNIX file system.
10. Read and understand specifications, scripts and programs.
11. Familiarize students with basic command of Git
12. Understand how to collaborate and work with Remote Repositories

PREREQUISITES : This course requires that the students are familiar with Commands and programming language such as C programming.

Module-1: Introduction to Shell	**RBT Levels: L1, L2,L3**	**6 Hours**

What is shell scripting? Importance of shell scripting, Different types of shells, creating shell script, making shell scripting executable, Shell input & output, Pipes and Filters, echo, print, read

Module-2: System and File Management	**RBT Levels: L1, L3**	**6 Hours**

Stages of a Process, Types of Process Management, Commands Used to Manage Processes like 'ps','top','kill','nice'.'renice','pkill', 'jobs', 'fg',' bg', Job scheduling commands like 'cron', 'at', 'batch'.
Different types of Files, working with files, working with Directories, basic File Attributes - displaying, searching, comparing, accessing long files and Zip files.

Module-3: User Management	**RBT Levels: L1,L3**	**6 Hours**

Creating Users & Groups, Managing Users and Groups, Create a Group, Modify a Group, Delete a Group, , Create a Account, Modify Account , Delete a Account.

Module-4: Introduction to Git	**RBT Levels:L1,L2,L3**	**6 Hours**

Setting Up and Basic Commands
Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.
Creating and Managing Branches
Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master." Write the commands to stash your changes, switch branches, and then apply the stashed changes.
Collaboration and Remote Repositories

Clone a remote Git repository to your local machine. Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch. Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.		
Module-5: Git workflows	**RBT Levels:L3,L4**	**6 Hours**
Git Tags and Releases Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository. **Analysing and Changing Git History** Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message? Write the command to list all commits made by the author "John Doe" between "2023-01-01" and "2023-12-31." Write the command to display the last five commits in the repository's history.		

Suggested Learning Resources:	
Reference Books:	
1.	Unix Shell Programming, Prof: Yogesh HK
2.	Unix Shell Programming, Yashwant Kanetkar
3.	Version Control with Git, 3rd Edition, by Prem Kumar Ponuthorai, Jon Loeliger Released October 2022,
4.	Publisher(s): O'Reilly Media, Inc.
5.	Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, https://git
Web links and Video Lectures (e-Resources): 3. Shell Scripting Tutorial 4. Getting Started with Git	
Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning 1. Create Git Project: Personal Portfolio Website Objective: Create a Git repository for a personal portfolio website. The repository will include a simple HTML file and showcase basic Git operations such as initialization, staging, committing, branching, and merging.	

Course outcomes:	
CO1	Ability to understand scripts and programs. Know the basics of Unix concepts and commands.
CO2	Ability to evaluate the UNIX file system and apply Changes in file system.
CO3	Ability to analyze authorizations for different users
CO4	Ability to use the basics commands related to git repository., Create and manage the branches, Apply commands related to Collaboration and Remote Repositories
CO5	Use the commands related to Git Tags, Releases and advanced git operations. Analyse and change the git history.

CO-PO Mapping															
CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PS O1	PSO 2	PSO3
CO1	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO2	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO3	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO4	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3
CO5	3	2	2	2	1	-	-	-	-	-	-	-	3	3	3

High-3, Medium-2, Low-1

Course Title	Indian Knowledge Systems	Semester	IV
Course Code	CS273	**Duration of CIE**	2 hours
Total No. of Contact Hours	30	**Credits**	2
No. of Contact Hours/week	2:0:0:1	**Teaching Dept.**	CSE/ISE

Course objective is to:

1. Introduce students to the vast body of Indian Knowledge
2. Familiarize students with the Vedic Corpus
3. Make students aware of plurality of philosophical traditions in India and harmony among them.
4. Orient students to wisdom through ages as found in Puranas, Itihasa and other texts.
5. Inspire students by making them aware of early advances in Mathematics in India.

PREREQUISITES : None.
Exposure to Sanskrit is desirable.

Module-1: Introduction to the Indian Knowledge Systems	**RBT Levels:L1,L2**	**6 Hours**

Do we need Indian Knowledge Systems? Importance of ancient knowledge. Defining Indian Knowledge Systems. The IKS Corpus – A Classification Framework. Caturdasha-Vidyashtana. Historicity of IKS. Some unique aspects of IKS.

T1: 1.1-1.6

Module-2: Vedic Corpus	**RBT Levels:L1,L2**	**6 Hours**

Introduction to Vedas. The four Vedas. The Four Divisions of Each Veda, Vedangas, Vedic Life
T1: 2.1-2.5

Module-3: Indian Philosophical Systems	**RBT Levels:L1,L2**	**6 Hours**

Indian Philosophical Systems – Development and Features, Vedic schools of philosophy, Sankhya and Yoga schools of philosophy, Nyaya and Vaisheshika School of philosophy, Purva-Mimansa andVednata Schools of Philosophy, Non-Vedic philosophical systems

T1:3.1-3.6

Module-4: Wisdom through the Ages	**RBT Levels:L1,L2**	**6 Hours**

Puranas, Ithihasa as a Source of Wisdom, Ramayana, Mahabharata, Niti-Sastras, Subhashita

T3: 4.1-4.6

Module-5: Indian Mathematics	**RBT Levels:L1,L2,L3**	**6 Hours**

Unique aspects of Indian Mathematics, Great Mathematicians and their Contributions, Arithmetic,Geometry, Trigonometry, Algebra, Binary mathematics and combinatorial algorithms, magic squares

T1:8.1-8.8

Follow on Courses:
Resources for Advanced Learners: https://nptel.ac.in/courses/111101080

Suggested Learning Resources:	
1.	Introduction to Indian Knowledge System, Concepts and Applications, B.Mahadevan, Vinayak Rajat Bhat, Nagendra Pavana R.N., Prentice-Hall India, 2022
Web links and Video Lectures (e-Resources): **1.** https://onlinecourses.swayam2.ac.in/imb24_mg21/preview **2.** https://onlinecourses.swayam2.ac.in/imb23_mg53/preview	
Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning Presentation	

Course outcomes:	
CO1	Awareness of Indian Knowledge Systems and its vast scope.
CO2	Awareness of Vedic Corpus and ability to understand the components
CO3	Appreciation of pluralistic philosophical tradition in India
CO4	Awareness of repositories of wisdom in Indian tradition and how they flow into day-to-day lives.
CO5	Awareness and appreciation of early advances made in India in the field of mathematics

CO-PO Mapping															
CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	-	3	-	-	-	-	-	3	-	-	-	3	-	-	-
CO2	-	3	-	-	-	-	-	3	-	-	-	3	-	-	-
CO3	-	3	-	-	-	-	-	3	-	-	-	3	-	-	-
CO4	-	3	-	-	-	-	-	3	-	-	-	3	-	-	-
CO5	-	3	-	-	-	-	-	3	-	-	-	3	-	-	-

High-3, Medium-2, Low-1

Arduino-Programming and Interfacing			
Course Code	CS274	**Semester**	IV
L:T:P:S	0:0:2:0	**CIE Marks**	50
Hrs. / Week	2	**SEE Marks**	50
Credits	2	**Total Marks**	100
Exam Hours	03		

Course objectives: This laboratory course enables	
1.	To program Arduino for sensors and other devices.
2.	To learn interfacing of Arduino with other devices.
3.	To familiarize the students with the programming of Arduino for real time application.

Pgm. No.	**List of Experiments / Programs**	**Hours/Week**	**COs**
Design and Write a program for the following Arduino based System.			
1	Automatic detection of Gas (LPG) leakage.	2	CO1
2	Fire Detection using Arduino and Flame sensor.	2	CO1
3	Controlling Relay shield from Bluetooth enabled device.	2	CO2
4	Develop a Heart rate monitoring system.	2	CO2
5	Design a Touch switch using Arduino and Touch sensor.	2	CO1
6	Interface a speaker with Arduino.	2	CO2
7	IOT based humidity and temperature monitoring system.	2	CO1
8	Obstacle detection system.	2	CO1
9	Automatically controlling the switching of a house outer lighting using photo resistance.	2	CO1
10	Display a digital clock on the LCD.	2	CO1
Practice Programs			
1	Interface the Temperature sensor LM35 with Arduino.		
2	Interface 4x4 keypad with Arduino.		

Conduct of Practical Examination:

Experiment distribution: Students are allowed to pick one experiment.

Mark distribution: 100%

Course outcomes:

CO1: Able to understand the types, working and characteristics of different sensors, actuators and Transducers.

CO2: Able to interface different devices with Arduino board.

CO3: Able to perform real time application using Arduino.

CO-PO Mapping															
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	2	2	--	-	-	2	-	2	-	-	2	2	2
CO2	1	2	2	2	-	-	-	2	-	2	-	-	1	1	1
CO3	1	2	2	2	-	-	-	2	-	2	--		2	2	2

Course Title	Universal Human Values and Life Skills	**Semester**	IV
Course Code	HV257	**CIE**	50
Total No. of Contact Hours	30	**SEE**	50
No. of Contact Hours/Week	2:0:0	**Total**	100
Credits	2	**Exam. Duration**	3 Hours
Teaching Dept	Any Department		

Course objective is to:

This course is intended to:

- To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
- To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value-based living in a natural way.
- To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behaviour and mutually enriching interaction with Nature.
- This course is intended to provide a much-needed orientation input in value education to the young enquiring minds.

Module – 1	**RBT Level/s :L1, L2**	**06 Hours**
Introduction to Value Education :Introduction to Value Education :Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education) Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfil the Basic Human Aspirations		
Module – 2	**RBT Level/s :L1, L2**	**06 Hours**
Harmony in the Human Being : Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health.		
Module-3	**RBT Level/s :L1, L2**	**06 Hours**
Module-4	**RBT Level/s :L1, L2**	**06 Hours**
Harmony in the Nature/Existence : Understanding Harmony in the Nature, Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence		

Module-5	RBT Level/s :L1, L2	06 Hours

Implications of the Holistic Understanding – a Look at Professional Ethics :Natural Acceptance of Human Values, Definitiveness of (Ethical) Human Conduct, A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order, Competence in Professional Ethics Holistic Technologies, Production Systems and Management Models-Typical Case Studies, Strategies for Transition towards Value-based Life and Profession

Suggested Learning Resources:

Textbooks:

1	The Textbook A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978- 93-87034- 47-1
2	The Teacher‟s Manual for A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G

Reference Books:

1	Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amar kantak, 1999.
2	Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3	The Story of Stuff (Book).
4	The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
5	Small is Beautiful - E. F Schumacher
6	Slow is Beautiful - Cecile Andrews

Web links and Video Lectures (e-Resources):

Value Education websites,

- https://www.uhv.org.in/uhv-ii,
- http://uhv.ac.in,
- http://www.uptu.ac.in
- Story of Stuff,
- http://www.storyofstuff.com
- Al Gore, An Inconvenient Truth, Paramount Classics, USA
- Charlie Chaplin, Modern Times, United Artists, USA
- IIT Delhi, Modern Technology – the Untold Story
- Gandhi A., Right Here Right Now, Cyclewala Productions
- https://www.youtube.com/channel/UCQxWr5QB_eZUnwxSwxXEkQw
- https://fdp-si.aicte-india.org/8dayUHV_download.php
- https://www.youtube.com/watch?v=8ovkLRYXIjE
- https://www.youtube.com/watch?v=OgdNx0X923I
- https://www.youtube.com/watch?v=nGRcbRpvGoU
- https://www.youtube.com/watch?v=sDxGXOgYEKM

Activity-Based Learning (Suggested Activities in Class)/ Practical Based learning

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

1. The methodology of this course is explorational and thus universally adaptable. It involves a systematic and rational study of the human being vis-à-vis the rest of existence.
2. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the activities will develop students' theoretical and applied skills.
3. State the need for UHV activities and its present relevance in the society and Provide real-life examples.
4. Support and guide the students for self-study activities. 5. You will also be responsible for

assigning homework, grading assignments and quizzes, and documenting students' progress in real activities in the field.

5. This process of self-exploration takes the form of a dialogue between the teacher and the students to begin with, and then to continue within the student in every activity, leading to continuous self evolution.
6. Encourage the students for group work to improve their creative and analytical skills.

Course outcomes:

Course outcome (Course Skill Set) At the end of the course, students are expected to become more aware of themselves, and their surroundings (family, society, nature);

Expected to positively impact common graduate attributes like:

1. Ethical human conduct
2. Socially responsible behaviour
3. Holistic vision of life
4. Environmentally responsible work
5. Having Competence and Capabilities for Maintaining Health and Hygiene
6. Appreciation and aspiration for excellence (merit) and gratitude for all

CO1	They would become more responsible in life, and in handling problems with sustainable solutions, while keeping human relationships and human nature in mind.
CO2	They would have better critical ability
CO3	They would also become sensitive to their commitment towards what they have understood (human values, human relationship and human society).
CO4	It is hoped that they would be able to apply what they have learnt to their own self in different day-to-day settings in real life, at least a beginning would be made in this direction.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1			1	1					1
CO2			1			1	1					1
CO3			1			1	1					1
CO4			1			1	1					1

High-3, Medium-2, Low-1

Course Title	NSS - Phase 2	**Semester**	IV
Course Code	NS258	**CIE**	50
Total No. of Contact Hours	25 Hours	**SEE**	0
No. of Contact Hours/week	0:2:0	**Total**	50
Credits	0	**Exam. Duration**	-----
Teaching Dept.	Any Department		

Course objective is to: National Service Scheme (NSS) will enable the students to:

1.Understand the community in which they work

2.Identify the needs and problems of the community and involve them in problem-solving

3.Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems

4.Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes

5.Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony

Module:1	**RBT Level/s :L1, L2**	**4 Hours**

Organic farming
Indian Agriculture (Past, Present and Future) Connectivity for marketing. Waste management– Public, Private and Govt organization, 5 R's.Plantation and adoption of plants. Know your plants. Awareness on Organic farming.

Module:2	**RBT Level/s :L1, L2**	**3 Hours**

Developing Water conservation techniques

To develop sustainable water management system, – Role of different stakeholders– Implementation Developing Sustainable Water management system for rural areas and implementation approaches.

Module: 3	**RBT Level/s :L1, L2**	**8 Hours**

Activity Based Programmes:
A. Campus Activities: Celebration of national importance days

Awareness Programmes – Preparing an actionable business proposal for enhancing the village income and approach for implementation. Importance of health, hygiene, and sanitation Healthy life style, HIV /AIDS, drugs and substance		
Module: 4	**RBT Level/s :L1, L2**	**10 Hours**
Off Campus Activities: Govt. school Rejuvenation and helping them to achieve good infrastructure and results, Women Empowerment Programme, Health Camps, Blood grouping awareness and Blood donation, Legal awareness Programme, Literacy Programme, Water Conservation Programme, One Day Special Camp in a village (preferably in adopted village.		

Course outcomes: **At the completion of the course. The student shall be able to**	
CO1	Describe the concept of Youth and compare the international definitions of the term Youth.
CO2	Students will be able to appreciate our demographic advantage and its role in nation building.
CO3	Know the growth and evolution of NSS and its role in Nation building through community service
CO4	Visualize the signs, symbols, logo of NSS and understand their broader meaning.

CO-PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1						1	1					1
CO2						1	1					1
CO3						1	1					1
CO4						1	1					1

High-3, Medium-2, Low-1

Course Title	Physical Education (PE) (Sports and Athletics) – Phase 2	**Semester**	IV
Course Code	PE258	**CIE**	50
Total No. of Contact Hours	25 Hours	**SEE**	0
No. of Contact Hours/Week	0:2:0	**Total**	50
Credits	0	**Exam. Duration**	------
Teaching Dept	Sports		

Course objective:

Physical education aims to develop all aspects of the human personality through physical and sports activities. Being a subject of science, it has its own value in society and human life. Physical Education is a form of one of the most effective means of education imparted through exercises, fun activities and sports. It is an integral part of the education system. It caters to the need for development of the students on physical, mental and social aspects.

Module – 1	**RBT Level/s :L1, L2**	**5 Hours**

Ethics and Moral Values:

A. Ethics in Sports.
B. Moral Values in Sports and Games

Module – 2	**RBT Level/s :L1, L2**	**15 Hours**

Specific Games (Any one to be selected by the student)

A. Volleyball – Attack, Block, Service, Upper Hand Pass and Lower hand Pass.
B. Throw ball – Service, Receive, Spin attack, Net Drop & Jump throw.
C. Kabaddi – Hand touch, Toe Touch, Thigh Hold, Ankle hold and Bonus.
D. Kho-Kho – Giving Kho, Single Chain, Pole dive, Pole turning, 3-6 up.
E. Table Tennis – Service (Fore Hand & Back Hand), Receive (Fore Hand & BackHand), Smash.
F. Athletics (Track / Field Events) – Any event as per availability of Ground.

.

Module - 3	**RBT Level/s :L1, L2**	**5 Hours**

Role of Organization and administration

Course outcomes: At the end of the course, the student will be able to	
CO1	Understand the ethics and moral values in sports and athletics.
CO2	Perform in the selected sports or athletics of student's choice
CO3	Understand the roles and responsibilities of organization and administrationof sports and games

CO-PO Mapping												
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1			1	1					1
CO2			1			1	1					1
CO3			1			1	1					1

High-3, Medium-2, Low-1

Course Title	Yoga for a Better Life - Phase 2	**Semester**	IV
Course Code	YG258	**CIE**	50
Total No. of Contact Hours	25	**SEE**	0
No. of Contact Hours/week	0:2:0	**Total**	50
Credits	0	**Exam. Duration**	-----
Teaching Dept			

Course objectives:

1) To enable the student to have good health.
2) To practice mental hygiene.
3) To possess emotional stability.
4) To integrate moral values.
5) To attain higher level of consciousness.

SEMESTER IV	**RBT Level/s :L1, L2**	**25 Hours**

Patanjali's Ashtanga Yoga, its need and importance. Yama :Ahimsa, satya, asteya, brahmacarya, aparigraha Niyama :shoucha, santosh, tapa, svaadhyaya, Eshvarapranidhan.
Suryanamaskar12 count- 4 rounds of practice
Asana, Need, importance of Asana. Different types of asana. Asana its meaning by name, technique, precautionary measures and benefits of each asana Different types of Asanas

a. Sitting
1. Sukhasana
2. Paschimottanasana

b. Standing
1. Ardhakati Chakrasana
2. Parshva Chakrasana

c. Prone line
1. Dhanurasana

d. Supine line
1. Halasana
2. Karna Peedasana

Meaning, importance and benefits of Kapalabhati. 40 strokes/min 3 rounds Meaning, Need, importance of Pranayama.
Different types. Meaning by name, technique, precautionary measures and benefits of each Pranayama Pranayama –
1. Suryanuloma –Viloma 2. Chandranuloma-Viloma 3. Suryabhedana 4. Chandra Bhedana 5. Nadishodhana

Suggested Learning Resources:
Textbooks:
1. Yogapravesha in Kannada by Ajitkumar
2. Light on Yoga by BKS Iyengar
3. Teaching Methods for Yogic practices by Dr. M L Gharote & Dr. S K Ganguly
4. Yoga Instructor Course hand book published by SVYASA University, Bengaluru

5. Yoga for Children –step by step – by Yamini Muthanna

Web links and Video Lectures (e-Resources):

Refer links

- **https://youtu.be/KB-TYlgd1wE**
- **https://youtu.be/aa-TG0Wg1Ls**

The Health Benefits of Yoga

The benefits of various yoga techniques have been supposed to improve

- body flexibility,
- performance,
- stress reduction,
- attainment of inner peace, and
- self-realization.

The system has been advocated as a complementary treatment to aid the healing of several ailments such as;

- coronary heart disease,
- depression,
- anxiety disorders,
- asthma, and
- extensive rehabilitation for disorders including musculoskeletal problems and traumatic brain injury.

The system has also been suggested as behavioral therapy for smoking cessation and substance abuse (including alcohol abuse). If you practice yoga, you may receive these physical, mental, and spiritual benefits:

- **Physical**
  1. Improved body flexibility and balance
  2. Improved cardiovascular endurance (stronger heart)
  3. Improved digestion
  4. Improved abdominal strength
  5. Enhanced overall muscular strength
  6. Relaxation of muscular strains
  7. Weight control
  8. Increased energy levels
  9. Enhanced immune system

- **Mental**
  1. Relief of stress resulting from the control of emotions
  2. Prevention and relief from stress-related disorders
  3. Intellectual enhancement, leading to improved decision-making skills
- **Spiritual**
  1. Life with meaning, purpose, and direction
  2. Inner peace and tranquility
  3. Contentment

Course outcomes: **At the completion of the course. The student shall be able to**	
CO1	Understand the meaning, aim and objectives of Yoga.
CO2	Perform Suryanamaskar and able to Teach its benefits.
CO3	Understand and teach different types of Pranayama, Asanas by name, its importance, methods

	and benefits.
CO4	Instruct Kapalabhati and its need and importance.
CO5	Coach different types of Kriyas , method to follow and usefulness.

CO-PO Mapping												
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1			1	1					1
CO2			1			1	1					1
CO3			1			1	1					1
CO4			1			1	1					1
CO5			1			1	1					1

High-3, Medium-2, Low-1

Course Title	Social Connect and Responsibilities	**Semester**	IV
Course Code	SC259	**CIE**	50
Total No. of Contact Hours	25	**SEE**	0
No. of Contact Hours/week	0:0:2	**Total**	50
Credits	0	**Exam. Duration**	-----
Teaching Dept	Any Department		

Course objectives:

- Provide a formal platform for students to communicate and connect to the surrounding.
- Create a responsible connection with the society.
- Understand the community in general in which they work.
- Identify the needs and problems of the community and involve them in problem –solving.
- Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems.
- Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.

General Instructions - Pedagogy :

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

1. In addition to the traditional lecture method, different types of innovative teaching methods may be adopted so that the activities will develop students' theoretical and applied social and cultural skills.
2. State the need for activities and its present relevance in the society and Provide real-life examples.
3. Support and guide the students for self-planned activities.
4. You will also be responsible for assigning homework, grading assignments and quizzes, and documenting students' progress in real activities in the field.
5. Encourage the students for group work to improve their creative and analytical skills.

SEMESTER IV	**RBT Level/s :L1, L2,L3**	**25 Hours**

Part I:

Plantation and adoption of a tree:

Plantation of a tree that will be adopted for four years by a group of BE / B.Tech students. (ONE STUDENT ONE TREE) They will also make an excerpt either as a documentary or a photo blog describing the plant's origin, its usage in daily life, its appearance in folklore and literature - – Objectives, Visit, case study, report, outcomes.

Part II :

Heritage walk and crafts corner:

Heritage tour, knowing the history and culture of the city, connecting to people around through their history, knowing the city and its craftsman, photo blog and documentary on evolution and practice of various craft forms - – Objectives, Visit, case study, report, outcomes.

Part III :

Organic farming and waste management:

Usefulness of organic farming, wet waste management in neighboring villages, and implementation in the campus – Objectives, Visit, case study, report, outcomes.

Part IV:

Water conservation:

Knowing the present practices in the surrounding villages and implementation in the campus, documentary or photo blog presenting the current practices – Objectives, Visit, case study, report, outcomes.

Part V :

Food walk:

City's culinary practices, food lore, and indigenous materials of the region used in cooking – Objectives, Visit, case study, report, outcomes

Activities:

Jamming session, open mic, and poetry: Platform to connect to others. Share the stories with others. Share the experience of Social Connect. Exhibit the talent like playing instruments, singing, one-act play, art-painting, and fine art.

PEDAGOGY:

The pedagogy will include interactive lectures, inspiring guest talks, field visits, social immersion, and a course project. Applying and synthesizing information from these sources to define the social problem to address and take up the solution as the course project, with your group. Social immersion with NGOs/social sections will be a key part of the course. Will all lead to the course project that will address the needs of the social sector?

COURSE TOPICS:

The course will introduce social context and various players in the social space, and present approaches to discovering and understanding social needs. Social immersion and inspiring conversional will culminate in developing an actual, idea for problem-based intervention, based on an in-depth understanding of a key social problem.

Course outcomes: **At the completion of the course. The student shall be able to**	
CO1	Communicate and connect to the surrounding.
CO2	Create a responsible connection with the society.
CO3	Involve in the community in general in which they work.
CO4	Notice the needs and problems of the community and involve them in problem –solving.
CO5	Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems.
CO6	Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.

Duration:

A total of 40 - 50 hrs engagement per semester is required for the 3rd semester of the B.E. /B.Tech. Program. The students will be divided into groups. Each group will be handled by faculty mentor. Faculty mentor will design the activities (particularly Jamming sessions open mic and poetry) Faculty mentors have to design the evaluation system as per VTU guidelines of scheme & syllabus.

Pedagogy – Guidelines :

It may differ depending on local resources available for the study as well as environment and climatic differences, location and time of execution.

Sl No	Topic	Group size	Location	Activity execution	Reporting	Evaluation Of the Topic
1.	**Plantation and adoption of a tree:**	May be individual or team	Farmers land/ parks / Villages / roadside/ community area / College campus etc…..	Site selection /proper consultation/Continuous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by Faculty
2.	**Heritage walk and crafts corner:**	May be individual or team	Temples / monumental places / Villages/ City Areas / Grama panchayat/ public associations/Government Schemes officers/ campus etc…..	Site selection /proper consultation/Continuous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by Faculty
3.	**Organic farming and waste management:**	May be individual or team	Farmers land / parks / Villages visits / Roadside/ community area / College campus etc…..	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by Faculty
4.	**Water conservation: & conservation techniques**	May be individual or team	Villages/ City Areas / Grama Panchayat/ public associations/Government Schemes officers / campus etc…..	site selection / proper consultation/Continuous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by Faculty
5.	**Food walk: Practices in society**	May be individual or team	Villages/ City Areas / Grama Panchayat/ public associations/Government Schemes officers/ campus etc…..	Group selection / proper consultation / Continuous monitoring / Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by Faculty

Plan of Action (Execution of Activities)	
Sl.NO	Practice Session Description
1	Lecture session in field to start activities
2	Students Presentation on Ideas
3	Commencement of activity and its progress
4	Execution of Activity
5	Execution of Activity
6	Execution of Activity
7	Execution of Activity
8	Case study based Assessment, Individual performance
9	Sector/ Team wise study and its consolidation
10	Video based seminar for 10 minutes by each student At the end of semester with Report.

- Each student should do activities according to the scheme and syllabus.
- At the end of semester student performance has to be evaluated by the faculty for the assigned activity progress and its completion.
- At last consolidated report of all activities from 1st to 5th, compiled report should be submitted as per the instructions and scheme.

Assessment Details for CIE (both CIE and SEE)

Weightage	CIE – 100%
Field Visit, Plan, Discussion	10 Marks
Commencement of activities and its progress Case study based Assessment	20 Marks
Individual performance with report	20 Marks
Sector wise study & its consolidation 5*5 = 25	25 Marks
Total marks for the course	100 Marks

- Implementation strategies of the project (NSS work).
- The last report should be signed by NSS Officer, the HOD and principal.
- At last report should be evaluated by the NSS officer of the institute.
- Finally the consolidated marks sheet should be sent to the university and also to be made available at LIC visit.

For each activity, 20 marks CIE will be evaluated for IA marks at the end of semester, Report and assessment copy should be made available in the department.

Students should present the progress of the activities as per the schedule in the prescribed practical session in the field. There should be positive progress in the vertical order for the benefit of society in general through activities.

CO-PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1			1			1	1					1
CO2			1			1	1					1
CO3			1			1	1					1
CO4			1			1	1					1
CO5			1			1	1					1

High-3, Medium-2, Low-1