BE SCHEME & SYLLABUS

Second Year
(III and IV Semester)

With effect from 2022-23

Computer Science & Engineering (Data Science)

ST JOSEPH ENGINEERING COLLEGE
AN AUTONOMOUS INSTITUTION
Vamanjoor, Mangaluru - 575028

MOTTO

Service & Excellence

VISION

To be a global premier Institution of professional education and research.

MISSION

- Provide opportunities to deserving students of all communities, the Christian students in particular for quality professional education.
- Design and deliver curricula to meet the national and global changing needs through student-centric learning methodologies.
- Attract, nurture and retain the best faculty and technical manpower.
- Consolidate the state-of-art infrastructure and equipment for teaching and research activities.
- Promote all round personality development of the students through interaction with alumni, academia and industry.
- Strengthen the Educational Social Responsibilities (ESR) of the institution.

ST JOSEPH ENGINEERING COLLEGE

An Autonomous Institution
Vamanjoor, Mangaluru - 575028

Affiliated to VTU – Belagavi & Recognized by AICTE New Delhi
NBA – Accredited: B.E.(CSE, ECE, EEE, ME and CIV) & MBA
NAAC – Accredited with A+

B.E. SCHEME & SYLLABUS

(With effect from 2022-23)

Computer Science & Engineering (Data Science)

SECOND YEAR

(III and IV Semester)

AUTONOMY AND ACCREDITATION

St Joseph Engineering College (SJEC) is an Autonomous Institute under Visvesvaraya Technological University (VTU), Belagavi, Karnataka State, and is recognized by the All-India Council for Technical Education (AICTE), New Delhi. SJEC is registered under the trust "Diocese of Mangalore, Social Action Department".

The SJEC has been conferred Fresh Autonomous Status from the Academic Year 2021-22. The college was granted autonomy by the University Grants Commission (UGC) under the UGC Scheme for Autonomous Colleges 2018 and conferred by VTU. The UGC Expert Team had visited the college on 28-29 November 2021 and rigorously assessed the college on multiple parameters. The fact that only a handful of engineering colleges in the state have attained Autonomous Status adds to the college's credibility that has been on a constant upswing. Autonomy will make it convenient for the college to design curricula by recognizing the needs of the industry, offering elective courses of choice and conducting the continuous assessment of its students.

At SJEC, the Outcome-Based Education (OBE) system has been implemented since 2011. Owing to OBE practised at the college, SJEC has already been accredited by the National Board of Accreditation (NBA). Five of the UG programs, namely Computer Science & Engineering, Mechanical Engineering, Electronics and Communication Engineering, Electrical & Electronics Engineering and Civil Engineering and MBA programs, have accreditation from the NBA.

Also, SJEC has been awarded the prestigious A+ grade by the National Assessment and Accreditation Council (NAAC) for five years. With a Cumulative Grade Point Average (CGPA) of 3.39 on a 4-point scale, SJEC has joined the elite list of colleges accredited with an A+ grade by NAAC in its first cycle. The fact that only a small percentage of the Higher Education Institutions in India have bagged A+ or higher grades by NAAC adds to the college's credibility that has been on a constant upswing.

The college is committed to offering quality education to all its students, and the accreditation by NAAC and NBA reassures this fact. True to its motto of "Service and Excellence", the college's hard work has resulted in getting this recognition, which has endorsed the academic framework and policies that the college has been practicing since its inception. The college has been leveraging a flexible choice-based academic model that gives students the freedom to undergo learning in respective disciplines and a transparent and continuous evaluation process that helps in their holistic development.

CONTENTS

ABOUT THE DEPARTMENT

Computer Science and Data Science are two interrelated fields that have become increasingly important in today's digital age. While Computer Science focuses on the study of computers and computational systems, Data Science is concerned with the extraction, analysis, and interpretation of complex data sets. Data Science has emerged with the growth of data and involves collecting, cleaning, and analysing large data sets using statistical techniques and machine learning algorithms to identify patterns and trends for better decision-making.

Combining Computer Science and Data Science (CSDS) will play an essential role in the digital age. As more and more data are generated, the demand for skilled professionals in these fields is only going to increase. As a result, students and professionals need to gain knowledge and skills in both Computer Science and Data Science to stay competitive in today's job market such as Business Intelligence Developers, Research Scientists, Big Data Engineer/ Architect, Software Engineer, Data Analyst, Data Scientist, Data Mining & Analysis, NLP Engineer, AI Engineer, Cyber Analyst, and Product manager that are highly demanding. Demand for data science engineers is expected to grow to 27.9% by 2026.

DEPARTMENT VISION

To impart value-based quality education with the motive of transforming mankind with excellence and competing areas of engineering, technology and management.

DEPARTMENT MISSION

1. Focus on the practical aspects of the curriculum to make learning a meaningful and interesting experience.
2. Encourage active collaboration with industries, communities, and fellow institutions within the country and abroad.
3. Infuse strong moral and ethical principles in students in order to make them conscientious citizens and excellent human beings.
4. Cultivate the competitive spirit required for success.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

1. To provide students with a solid foundation and the ability to use engineering concepts, mathematics, physics, and humanities required to develop, analyse, design, and implement solutions to the problems in intelligent computing and business systems.
2. To develop in students, the knowledge of computer science and engineering to work in domains such as artificial intelligence, machine learning and data science.
3. To foster in students, the capacity of teamwork through efficient communication in multidisciplinary projects.
4. To prepare students for building successful careers in artificial intelligence, data science and business systems to meet the needs of society while incorporating professional ethics.
5. To inspire learners to pursue higher education in their desired fields and engage in research.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations on complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and the synthesis of information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

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

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

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

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

12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

1. **Entrepreneurship and Freelancing**: Recognize the tenets of entrepreneurship, freelancing and the prerequisites for starting a business in the IT or related fields.

2. **Competitive Exams**: Participate skillfully in competitive examinations for certification, professional advancement, and admission to higher studies.

III Semester (B.E. - CSDS)													
SI. No	**Course and Course Code**		**Course Title**	**Teaching Department**	**Paper Setting Board**	**Teaching Hours/Week**			**Examination**				**Credits**
						Theory Lecture	**Tutorial**	**Practical/ Drawing**	**Duration in hours**	**CIE Marks**	**SEE Marks**	**Total Marks**	
						L	**T**	**P**					
1	PCC	22CDS31	Mathematics for Information Technology	MAT	MAT	2	2	-	03	50	50	100	3
2	IPCC	22CDS32	Digital Principles and Design (Integrated)	CSDS	CSDS	2	2	2	03	50	50	100	4
3	IPCC	22CDS33	Data Structures and Applications (Integrated)	CSDS	CSDS	3	-	2	03	50	50	100	4
4	PCC	22CDS34	Computer Organization and Architecture	CSDS	CSDS	3	-	-	03	50	50	100	3
5	ESC	22CDS35X	ESC/ETC/PLC	CSDS	CSDS	3	-	-	03	50	50	100	3
6	PCCL	22CDS36L	Object Oriented Programming with Java Lab	CSDS	CSDS	-	-	2	03	50	50	100	1
7	HSMC	22UHV37	Universal Human Values - II	COM	COM	2	-	-	02	50	50	100	2
		22BFE37	Biology for Engineers	COM	COM								
8	AEC/SDC	22IEP38	IoT Enabled Prototyping	COM	COM	-	-	2	02	50	50	100	1
9	MNCC	22ITB39A / 22ITC39B	Industry Oriented Training – Business Etiquettes/ Industry Oriented Training – Computing Skills	COM	COM	-	-	2	02	50	-	50	-
					Total	15	4	10	24	**450**	**400**	**850**	**21**

22CDS35X : Engineering Science Course/Emerging Technology Course/Programming Language Course			
22CDS351	Introduction to R Programming	22CDS352	Microcontroller and Embedded Systems

IV Semester (B.E. - CSDS)

SI. No	Course and Course Code		Course Title	Teaching Department	Paper Setting Board	Teaching Hours/Week			Examination				Credits
						Theory Lecture	Tutorial	Practical/ Drawing	Duration in hours	CIE Marks	SEE Marks	Total Marks	
						L	T	P					
1	PCC	22CDS41	Linear Algebra and Statistical Methods	MAT	MAT	2	2	-	03	50	50	100	3
2	IPCC	22CDS42	Design and Analysis of Algorithms (Integrated)	CSDS	CSDS	2	2	2	03	50	50	100	4
3	IPCC	22CDS43	Database Management System (Integrated)	CSDS	CSDS	3	-	2	03	50	50	100	4
4	PCC	22CDS44	Operating System	CSDS	CSDS	3	-	-	03	50	50	100	3
5	ESC	22CDS45X	ESC/ETC/PLC	CSDS	CSDS	3	-	-	03	50	50	100	3
6	PCCL	22CDS46L	Application Development using Python Laboratory	CSDS	CSDS	-	-	2	03	50	50	100	1
7	HSMC	22UHV47	Universal Human Values – II	COM	COM	2	-	-	02	50	50	100	2
		22BFE47	Biology for Engineers	COM	COM								
8	AEC/SDC	22CTE48	Computational Tools for Engineers	COM	COM	-	-	2	02	50	50	100	1
9	AEC/SDC	22ITB49A / 22ITC49B	Industry Oriented Training – Business Etiquettes/ Industry Oriented Training – Computing Skills	COM	COM	-	-	2	02	50	-	50	-
					Total	15	4	10	24	**450**	**400**	**850**	**21**

22CDS45X : Engineering Science Course/Emerging Technology Course/Programming Language Course			
22CDS451	C# Programming with Dot Net	22CDS452	Systems Programming

III Semester

Mathematics for Information Technology

Course Code	22CDS31	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	2:2:0	SEE	3 Hours
Total Hours	40	Credits	03

Course Learning Objectives:

1. Demonstrate the real-life application of Linear Programming
2. Apply propositional logic in knowledge representation
3. Use graph theory in computer applications

Module-1: **8 hours**

Linear Programming (LP) : Linear Programming Problems (L.P.P): Graphical method, General Linear programming Problem, Canonical and standard forms of L.P.P. Basic solution, Basic feasible solution, Optimal solution, Simplex Method-Problems. Artificial variable technique.

Module-2: **8 hours**

Fundamentals of Logic: Basic Connectives and Truth Tables, Logical Equivalence: The laws of logic, Rules of inference. Open Statement, Quantifiers

Module-3: **8 hours**

Enumeration and Generating Functions: Inclusion-exclusion principle, rook polynomials. First order linear recurrence relation, Second order linear homogeneous recurrence relations with constant coefficients

Module-4: **8 hours**

Graph Theory – I:

Definition and examples of Graphs, Subgraphs and Isomorphism. Vertex Degree and Hand Shaking Property. Walks and their classification, Euler Trails and Circuits.

Module-5: **8 hours**

Graph Theory – II:

Trees – Definitions, properties and examples. Rooted Trees.

Dijkstra's Shortest-Path Algorithm. Minimal Spanning Trees: The Algorithms of Kruskal and Prim.

Course Outcomes: At the end of the course the student will be able to:

22CDS31.1	Analyze and solve linear programming models of real-life situations and solve LPP by the simplex method
22CDS31.2	Use propositional and predicate logic in knowledge representation and truth verification
22CDS31.3	Apply inclusion principle, rook polynomial in computer applications.
22CDS31.4	Apply the concepts of different types of graphs in Computer Science Engineering
22CDS31.5	Explain the concept of Trees
22CDS31.6	Find the shortest path using Dijkstra's algorithm

Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Higher Engineering Mathematics	B.S Grewal	Khana Publishers	44th Edition 2021
2	Discrete and Combinatorial Mathematics	Ralph P. Grimaldi and B V Ramana	Pearson Education, Asia	5th Edition 2017
3	A first look at Graph Theory	John Clark and D. A. Holton	World Scientific Publishers	2nd Edition 2011
4	Graph Theory with applications to engineering and computer science	Narasingh Deo	Prentice -Hall India	25th Edition 2003
Reference Books				
1	Discrete Mathematical Structures with Applications to Computer Science	J.P. Tremblay and R. Manohar	Tata – McGraw Hill Publications	1st Edition, 2017
2	Discrete Mathematics and its Applications	Kenneth H. Rosen	Tata – McGraw Hill Publications	7th Edition, 2017
3	Topics in Algebra	I N Herstein	Wiley Eastern Limited	2nd Edition, 1993

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=E40r8DWgG40&list=PLEAYkSg4uSQ2fXcfrTGZdPuTmv98bnFY5
- https://www.youtube.com/watch?v=NR0qG64gZUs
- https://www.youtube.com/watch?v=nKsC70MtzkY
- https://www.youtube.com/watch?v=Ic5PHlKXC-s&list=PLgMDNELGJ1Ca7hpEIYtWvMXKcTx88OD2O&index=4
- https://www.youtube.com/watch?v=a2QgdDk4Xjw&list=PLAD23E7AEFE221F70

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CDS31.1	2		1											
22CDS31.2	2	1												
22CDS31.3	2		1											
22CDS31.4		2	1											
22CDS31.5	2											1		
22CDS31.6		2	1											

1: Low 2: Medium 3: High

Digital Principles and Design (Integrated)

Course Code	**22CDS32**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:2:2)	SEE Marks	50
Credits	**04**	Exam Hours	03

Course Learning Objectives:

1. Make use of basic gates and design the logic circuits.
2. Apply the simplifying techniques in the design of combinational circuits
3. Differentiate the combinational and sequential circuits
4. Demonstrate the use of flip-flops in the construction of registers and counters
5. Illustrate how to write simple HDL programs which describe the digital circuits

Module-1 **8 Hours**

Digital Logic and Principles of Combination Logic : Review of Basic gates, Universal gates, Positive and Negative logic , Boolean Laws and theorems, minimization of completely and incompletely specified switching functions, Simplifying Max term equations, Sum of product method, Product of sums method, Product of sums simplification

Module-2 **8 Hours**

Combinational Logic Circuit Design:, Karnaugh map simplification, Don't care conditions, Simplification by Quine McCluskey method, Determination of Prime implicants, Simplification using map-entered variables, Gate delays and Timing diagrams, Hazard and Hazard covers, Introduction to HDL: Verilog HDL, Describing input/output, writing module body, HDL Implementation models.

Module-3 **8 Hours**

Data Processing Circuits : Multiplexers, De-multiplexers, decoder, BCD to Decimal decoder, seven segment decoder, encoders, Ex-OR gates, Parity generators and checkers, Magnitude comparators, Read only memory, Programmable array logic(PAL), Programmable logic arrays(PLA) HDL implementation: HDL of data processing circuits, Arithmetic circuits using HDL

Module-4 **8 Hours**

Latches and Flip-Flops: RS Flip-Flop, Gated Flip-Flops: Clocked RS and D Flip-Flops, Edge triggered RS Flip-Flops, Edge triggered D Flip-Flops, Edge triggered JK Flip-Flops, JK master slave Flip-Flop, switch contact bounce circuits, various representation of Flip- Flops, Analysis of sequential circuits.

Module-5 **8 Hours**

Registers and Counters: Registers: Types of registers, Application of shift registers HDL implementation: HDL implementation of Flip-Flops and registers, Asynchronous counters, Decoding gates, Synchronous counters, changing the counter modulus, decade counters, presettable counters, Sequential Parity checker.

List of Laboratory Experiments related to above modules – 2 hours each

1. Design and implementation of a Half adder, Half Substractor and a Full Adder using basic gates. Implement Full Substractor in Verilog HDL.
2. Given a 4-variable logic expression, simplify it using appropriate technique and realize the simplified logic expression using 8:1 multiplexer IC and implement the same in Verilog HDL
3. Realize a J-K Master / Slave Flip-Flop using NAND gates and verify its truth table. And implement JK flip flop in Verilog HDL
4. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.
5. Design and implement an asynchronous counter using decade counter IC to count up

from 0 to n (n<=9) and demonstrate on a 7-segment display (using IC-7447).

6. Design and Testing Shift Register/Ring Counter/Johnson Counter

Open ended experiment covering the concept of entire syllabus

1. Design and Testing Sequence Generator
2. Use Universal gates and IC's for code conversion and arithmetic Operations

Course Outcomes: At the end of the course the student will be able to:	
22CDS32.1	Explain the usage of basic gates, universal gates and Boolean laws in designing various digital circuits
22CDS32.2	Apply the simplification techniques like Karnaugh map and Quine Mc-clusky to design various combinational circuits.
22CDS32.3	Describe the operation and design of various data processing circuits and implement multiplexers circuit.
22CDS32.4	Identify the various types of flip-flops and use them in the design of Registers and Counters
22CDS32.5	Differentiate between Moore and Mealy model and construct different types of counters using these models.
22CDS32.6	Develop Verilog HDL programs to implement simple combinational and sequential circuits

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Digital Principles and Application	Donald P Leach, Albert Paul Malvino & Goutam Saha	Tata McGraw Hill	8th Edition, 2017
2	Fundamentals of Logic Design	Charles H Roth and Larry L kinney	Cengage Learning	7th Edition, 2019
Reference Books				
1	Fundamentals of Digital Logic Design with VHDL	Stephen Brown, Zvonko Vranesic	Tata McGraw Hill	2nd Edition 2005
2	Illustrative Approach to Logic Design	R D Sudhake Samuel	Pearson Education	2010

Web links/Video Lectures/MOOCs/papers:

https://onlinecourses.nptel.ac.in/noc22_ee110/preview
https://cse15-iiith.vlabs.ac.in/2

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS32.1	3	1												1
22CDS32.2	1	2	2											1
22CDS32.3	3		1											1
22CDS32.4		1		2										1
22CDS32.5		1		2										1
22CDS32.6		1			2									1

1: Low 2: Medium 3: High

Data Structures and Applications

Course Code	**22CDS33**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Integrated	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	3:0:2	SEE	3 Hours
Total Hours	40 hours Theory + 10 Lab slots	Credits	04

Course Learning Objectives: The objective of the course is to

- Understand the fundamentals of data structures and applications that are essential for programming and problem solving.
- Learn linear and non-linear data structures.
- Implement various basic and advanced data structures operations.
- Understand various sorting and searching techniques.
- Know the importance of Hashing

Module-1: Basic Concepts & Strings **8 hours**

Basic Concepts: Data Structures, Classifications (Primitive & Non-Primitive), Data structure Operations, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions, Polynomials and Sparse Matrices in arrays.

Strings: Basic Terminology, Storing Operations and Pattern Matching algorithms.

TB1: **2.2,2.3,2.4,2.5,2.6** **TB2: 2.1,2.2,2.3**

Module-2: Stacks, Queues **8 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. Recursion.

Queues: Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues

TB1: Ch 3.1,3.2,3.3,3.4,3.6 **TB2: Ch 7.7,8.4**

Module-3: Linked Lists, Graphs **8 hours**

Linked Lists: Definition, Representation of linked lists in Memory, linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, header linked lists. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples
Graphs: Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search.
TB1: Ch 4.4,4.7 **TB2: Ch 6.1,6.2,6.3,6.4,6.5.6.6,13.1,13.2,13.3,13.5,13.6**

Module-4: Trees **8 hours**

Trees: Introduction, Binary Trees, Binary Tree Traversals, Additional Binary Tree Operations, Threaded Binary Trees, Binary Search Trees,
Forests, Multi-way Search Trees: Introduction, B Trees, B+ Trees.
TB1: Ch 5.1,5.2,5.3,5.4,5.5,5.7,5.9 **TB2: Ch 11.1,11.2,11.3**

Module-5: Searching, Sorting, Hashing **8 hours**

Searching and Sorting: Jump Search, Insertion sort, Radix Sort, Shell Sort.
Hashing and Collision: Introduction, Hash Tables, hash Functions, Different Hash Functions, Collisions, Pros and Cons of Hashing, Applications of Hashing

TB2: Ch 14.1,14.5,14.8,14.12,14.14,15.1,15.2,15.3,15.4,15.5,15.6,15.7

PRACTICAL MODULE

1. Design, Develop and Implement a menu driven Program for the following
   a) Demonstrate dynamic allocation of 2D array of integers (use suitable pointer)
   b) Read m×n sparse matrix into an array.
   c) Compute transpose of m×n sparse matrix using fast transpose algorithm
   d) Display sparse matrix.

   Support the program with functions for each of the above operations.
2. Design, Develop and Implement a Program for the following operations on Strings.
   a) Read a main String (STR) and a Pattern String (PAT)
   b) Implement KMP algorithm to Perform Pattern Matching Operation: Find the occurrences of PAT in STR. Report suitable messages in case PAT does not exist in STR.

   Support the program with functions for each of the above operations. Don't use Built-in functions.
3. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +(add), -(sub), *(multiple), /(division), %(Remainder), ^(Power) and alphanumeric operands.
4. Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size (MAX)

   a. Insert an Element on to Circular QUEUE
   b. Delete an Element from Circular QUEUE
   c. Demonstrate *Overflow* and *Underflow* situations on Circular QUEUE
   d. Display the status of Circular QUEUE
   e. Exit

   Support the program with appropriate functions for each of the above operations.
5. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: *USN, Name, Branch, Sem, PhNo*

   a. Create a SLL of N Students Data by using *front insertion*.
   b. Display the status of SLL and count the number of nodes in it
   c. Perform Insertion / Deletion at End of SLL
   d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack)
   e. Exit
6. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers

   a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
   b. Traverse the BST in Inorder, Preorder and Post Order
   c. Search the BST for a given element (KEY) and report the appropriate message
   d. Exit
7. Design, develop and Implement a Program to sort n elements using Radix Sort Algorithm.
8. Design, develop and Implement a Program for the following operations on Hash Table. 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.
9. Open ended experiment covering the concept of entire syllabus: Online shopping application

Course Outcomes: At the end of the course the student will be able to:	
22CDS33.1	Apply data structures (pointers, arrays, structures and strings) for data organization.
22CDS33.2	Make use of data structures such as Stacks, Queues for data organization.
22CDS33.3	Apply Linked Lists and Graphs for data representation, Insertion, Deletion and Search Operations.
22CDS33.4	Make use of Tree data structure for data ordering, data searching and evaluating expressions.
22CDS33.5	Analyze various Searching algorithms, Sorting algorithms and Hash table organization.
22CDS33.6	Analyze implementation of data structures to real life applications involving data storage, access and organization

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Fundamentals of Data Structures in C	Ellis Horowitz and Sartaj Sahni	Universities Press	2nd Edition, 2008.
2	Data Structures using C	Reema Thareja	Oxford Press	2nd Edition, 2014
Reference Books				
1	Data Structures using C	Aaron M. Tenenbaum, Yedidyah Langsam , Moshe, J.Augenstein	Pearson Education	1st Edition, 2019
2	Data Structures: A Pseudocode Approach with C	Richard F. Gilberg and Behrouz A. Forouzan	Cengage Learning	2nd Edition, 2005

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=CHhwJjR0mZA
- https://www.youtube.com/watch?v=xLetJpcjHS0&list=PLBlnK6fEyqRj9lld8sWIUNwlKfdUoPd1Y
- https://www.youtube.com/watch?v=B31LgI4Y4DQ
- https://masterraghu.com/subjects/Datastructures/ebooks/rema thareja.pdf
- https://archive.nptel.ac.in/courses/106/102/106102064/

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CDS33.1	1	2							2					1
22CDS33.2	1	2							2					1
22CDS33.3	1	2							2					1
22CDS33.4	1	2					1		2					1
22CDS33.5	1	2					1		2					
22CDS33.6		2					1	2	2	2				

1: Low 2: Medium 3: High

Computer Organization and Architecture			
Course Code	**22CDS34**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	**03**	Exam Hours	03

Course Learning Objectives:

1. Explain the basic subsystems of a computer, their organization, structure and operations
2. Illustrate the concept of programs as sequences of machine instructions
3. Demonstrate different ways of communicating with I/O devices and standard I/O interfaces
4. Describe arithmetic operations with integer operands.
5. Appraise the Computer Architecture and instruction level parallelism

Module-1: Structure & Machine Instructions

Basic Structure of Computers: Basic Operational Concepts, Bus Structures, Performance- Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement.

Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Subroutines and use of stack frames, Additional Instructions. TB1 **8 Hours**

Module-2: Input/Output Organization

Input/Output Organization: Basic Input and Output Operations, Accessing I/O Devices, Interrupts – Interrupt hardware, Enabling and disabling of interrupts, Handling multiple devices, Controlling device requests, exceptions, Interrupt Hardware, Direct Memory Access, Bus arbitration, Synchronous and Asynchronous bus. TB1 **8 Hours**

Module-3: Memory & Multiprocessors

Memory unit: Memory Hierarchy, Basics of Cache memory, Cache mapping techniques, **Multiprocessor Architecture:** Centralized shared-memory architecture, Distributed memory architecture, **Basic scheme for enforcing coherence:** Snooping coherence protocols, Basic implementation techniques. TB1 and TB2 **8 Hours**

Module-4: Arithmetic & Bus Organizations

Integer arithmetic: Numbers, Arithmetic operations and characters, Overflow in integer arithmetic, Multiplication of two numbers, Signed operand multiplication, Booth algorithm, Bit pair recoding and CSA, integer division. **Basic Processing unit:** Fundamental concepts, Execution of complete instruction, Multiple bus organization. TB1 **8 Hours**

Module-5: ILP & Pipelining

Instruction level parallelism (ILP): Introduction and challenges, Data dependencies and Hazards: Data dependencies, Name dependencies, Control Dependencies, Pipelining: Introduction, A simple implementation of a RISC instruction set, The classic five-stage pipeline for a RISC processor, Basic performance issues in pipelining. TB2 **8 Hours**

Course Outcomes: At the end of the course the student will be able to:	
22CDS34.1	Interpret the basic structure and operations of computers and demonstrate the significance of addressing modes and instruction sequencing.
22CDS34.2	Select the different ways of communicating with I/O devices and standard I/O interfaces.
22CDS34.3	Illustrate cache memory mapping techniques, various memory architectures and protocols for cache coherence.
22CDS34.4	Apply different algorithms to perform arithmetic operations.
22CDS34.5	Illustrate organization of a processor with single and multiple bus for instruction execution.

22CDS34.6	Appraise the importance of pipelining to achieve instruction level parallelism.

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Computer Organization	C Hamacher, Z Vranesic	Tata McGraw Hill	5th Edition 2011
2	Computer Architecture: A Quantitative Approach	John L Hennessy, David A Patterson	Elsevier	5th Edition 2012
Reference Books				
1	Computer Organization and Design	David A. Patterson, John L. Hennessy	M.K Publishers	4th Edition, 2010
2	Computer Organization and Architecture	William Stallings	Pearson	9th Edition, 2014
3	Computer Organization and Design MIPS Edition	Patterson	Elsevier;	6th Edition 2021
4	Computer Organization and Architecture	J. S. Katre, Harish G.Narula, Khushboo Shah	Tech Knowledge Publications	2020

Web links/Video Lectures/MOOCs/papers

1. https://www.coursera.org/learn/comparch
2. https://nptel.ac.in/courses/106103068
3. https://www.youtube.com/watch?v=leWKvuZVUE8&list=PL1A5A6AE8AFC187B7

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS34.1	2											2		
22CDS34.2			3	2										
22CDS34.3							2							2
22CDS34.4	3											2		
22CDS34.5						2						3		
22CDS34.6			1	2										

1: Low 2: Medium 3: High

Introduction to R Programming

Course Code	**22CDS351**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L: T:P)	3:0:0	SEE	3 Hours
Total Hours	40 Hours	Credits	03

Course Learning Objectives: The objective of the course is to

- Understand the basics of Fundamentals of R.
- Understands the loading, and retrieval techniques of data.
- Understand how data is analyzed and visualized using statistical functions.
- Understand how Data frames work.
- Work on built-in real-time cases for analysis and visualization

Module-1: Introduction **8 hours**

What is R? – Why R? – Advantages of R over Other Programming Languages - R Studio: R command Prompt, R script file, comments – Handling Packages in R: Installing an R Package, Few commands to get started: installed.packages(), package description (), help(), find.package(), library() - Input and Output – Entering Data from the keyboard – Printing fewer digits or more digits. Simple programs on R.

Module-2: Data Structures in R **8 hours**

R Data Types: Vectors, Lists, Matrices, Arrays, Factors, Data Frames. **R - Variables:** Variable assignment, Data types of Variable, Finding Variable ls(), Deleting Variables. **R Operators:** Arithmetic Operators, Relational Operators, Logical Operator, Assignment Operators, Miscellaneous Operators - **R Decision Making:** if statement, if – else statement, if – else if statement, switch statement – **R Loops**: repeat loop, while loop, for loop - Loop control statement: break statement, next statement.

Module-3: R Functions **8 hours**

R-Function: function definition, Built in functions: mean(), paste(), sum(), min(), max(), seq(), user-defined function, calling a function, calling a function without an argument, calling a function with argument values - **R-Strings** – Manipulating Text in Data: substr(), strsplit(), paste(), grep(), toupper(), tolower() - **R Vectors** – Sequence vector, rep function, vector access, vector names, vector math, vector recycling, vector element sorting - **R List** - Creating a List, List Tags and Values, Add/Delete Element to or from a List, Size of List, Merging Lists, Converting List to Vector - **R Matrices** – Accessing Elements of a Matrix, Matrix Computations: Addition, subtraction, Multiplication and Division- **R Arrays**: Naming Columns and Rows, Accessing Array Elements, Manipulating Array Elements, Calculation Across Array Elements - R Factors –creating factors, generating factor levels gl()

Module-4: Data Frames **8 hours**

Data Frames –Create Data Frame, Data Frame Access, Understanding Data in Data Frames: dim(), nrow(), ncol(), str(), Summary(), names(), head(), tail(), edit() functions - Extract Data from Data Frame, **Expand Data Frame:** Add Column, Add Row - Joining columns and rows in a Data frame rbind() and cbind() – Merging Data frames merge() – Melting and Casting data melt(), cast(). **Loading and handling Data in R:** Getting and Setting the Working Directory – getwd(), setwd(), dir() **- R-CSV Files** - Input as a CSV file, Reading a CSV File, Analyzing the CSV File: summary(), min(), max(), range(), mean(), median(), apply() - Writing into a CSV File **– R -Excel File** – Reading the Excel file

Module-5: R Analytics **8 hours**

Descriptive Statistics: Data Range, Frequencies, Mode, Mean and Median: Mean Applying Trim Option, Applying NA Option, Median - Mode - Standard Deviation – Correlation - **Spotting Problems in Data with Visualization**: visually Checking Distributions for a single Variable - R – Pie Charts: Pie Chart title and Colors – Slice Percentages and Chart Legend, 3D Pie Chart – R Histograms – Density Plot - R – Bar Charts: Bar Chart Labels, Title and Colors.

Course Outcomes: At the end of the course the student will be able to:	
22CDS351.1	Use the functionalities offered by R packages.
22CDS351.2	Apply fundamentals of R for a given problem.
22CDS351.3	Utilize different data structures and organize data using functions.
22CDS351.4	Demonstrate data handling by creating, manipulating, and analyzing
22CDS351.5	datasets using data frames or files.
22CDS351.6	Analyze descriptive statistics and produce data visualizations.

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Data Analytics using R	Seema Acharya	Mc Graw Hill	Data Analytics using R
Reference Books				
1	R Programming for Beginners	Sandip Rakshit	Mc Graw Hill Education	1st Edition, 2017
2	R for Dummies	Andrie de Vries, Joris Meys	A Wiley Brand	2nd Edition, 2015

Web links and Video Lectures (e-Resources):

- https://jrnold.github.io/r4ds-exercise-solutions/index.html
- https://www.r-project.org/
- https://cran.r-project.org/
- https://youtu.be/yZ0bV2Afkjc

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS351.1			2											
22CDS351.2		2									2	2		
22CDS351.3		2										1		1
22CDS351.4				2					1			2		2
22CDS351.5	2													
22CDS351.6	3				1									

1: Low 2: Medium 3: High

Microcontroller and Embedded Systems

Course Code	**22CDS352**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	3:0:0	SEE	3 Hours
Total Hours	40 Hours	Credits	03

Course Learning Objectives: The objective of the course is to

- Understand the fundamentals of ARM based systems, basic hardware components, selection
- methods and attributes of an embedded system.
- Program ARM controller using the various instructions
- Identify the applicability of the embedded system
- Comprehend the real time operating system used for the embedded system

Module-1: Introduction to Processors : 8 hours

Microprocessors versus Microcontrollers, ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software.

ARM Processor Fundamentals: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and the Vector Table , Core Extensions.

TB1 : Ch-1.1 to 1.4, 2.1 to 2.5

Module-2: Introduction to the ARM Instruction Set : 8 hours

Data Processing Instructions , Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants.

ARM programming using Assembly language: Writing Assembly code, Profiling and cycle counting, instruction scheduling, Register Allocation, Conditional Execution, Looping Constructs.

TB1 : Ch-3.1 to 3.6, 6.1 to 6.6

Module-3: Embedded System Components: 8 hours

Embedded Vs General computing system, History of embedded systems, Classification of Embedded systems, Major applications areas of embedded systems, purpose of embedded systems. Core of an Embedded System including all types of processor/controller, Memory, Sensors, Actuators, LED, 7 segment LED display, stepper motor, Keyboard, Push button switch, Communication Interface (onboard and external types), Embedded firmware, Other system components.

TB2 : Ch-1.2 to 1.6, 2.1 to 2.6

Module-4: Embedded System Design Concepts: 8 hours

Characteristics and Quality Attributes of Embedded Systems, Operational quality attributes, non-operational quality attributes, Embedded Systems-Application and Domain specific, Hardware Software Co-Design and Program Modelling, embedded firmware design and development.

TB2 : Ch-3, 4, 7.1, 7.2, 9.1, 9.2, 9.3.1, 9.3.2

Module-5: RTOS and IDE for Embedded System Design: 8 hours

Operating System basics, Types of operating systems, Task, process and threads (Only POSIX Threads with an example program), Thread preemption, Multiprocessing and Multitasking, Task Communication (without any program), Task synchronization issues – Racing and Deadlock, Concept of Binary and counting semaphores (Mutex example without any program), How to choose an RTOS, Integration and testing of Embedded hardware and firmware, Embedded system Development Environment – Block diagram (excluding Keil), Disassembler/decompiler, simulator, emulator and debugging techniques, target hardware debugging, boundary scan.

TB2 : Ch-10.1, 10.2, 10.3, 10.4 , 10.7, 10.8.1.1, 10.8.1.2, 10.8.2.2, 10.10, 12, 13.1, 13.3, 13.4, 13.5, 13.6

Course Outcomes: At the end of the course the student will be able to:

22CDS352.1	Describe the architectural features and instructions of ARM microcontroller

22CDS352.2	Apply the knowledge gained for Programming ARM for different applications.
22CDS352.3	Interface external devices and I/O with ARM microcontroller.
22CDS352.4	Interpret the basic hardware components and their selection method based on the characteristics and attributes of an embedded system.
22CDS352.5	Develop the hardware /software co-design and firmware design approaches.
22CDS352.6	Demonstrate the need of real time operating system for embedded system applications

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	ARM system developer's guide	Andrew N Sloss, Dominic Symes and Chris Wright	Morgan Kaufman publishers	No edition 2004
2.	Introduction to Embedded Systems	Shibu K V	Tata McGraw Hill Education	2nd Edition, 2016
Reference Books				
1	Microcontroller (ARM) and Embedded System	Raghunandan..G.H	Cengage learning Publication	1st Edition,2020
2	The Insider's Guide to the ARM7 Based Microcontrollers	Zachary Lasiuk, Pareena Verma, Jason Andrews	Hitex Ltd	1st Edition, 2005
3	ARM System-on-Chip Architecture,	Steve Furber	Pearson	2nd Edition, 2015
4.	Embedded System	Raj Kamal	Tata McGraw-Hill Publishers	2nd Edition, 2008

Web links and Video Lectures (e-Resources):

- https://ict.iitk.ac.in/courses/embedded-systems
- https://archive.nptel.ac.in/courses/106/105/106105193/
- http://fab.cba.mit.edu/classes/863.18/CBA/people/alex/week-7-embedded-programming/index.html
- https://www.coursera.org/learn/introduction-embedded-systems
- https://courseware.cutm.ac.in/courses/2285

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS352.1	2	1												1
22CDS352.2	1	2	2											1
22CDS352.3	2		1											1
22CDS352.4	1	2		2										1
22CDS352.5	1	2		2										1
22CDS352.6		2			2									1

1: Low 2: Medium 3: High

Object Oriented Programming with Java Lab

Course Code	22CDS36L	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	**01**	Exam Hours	03

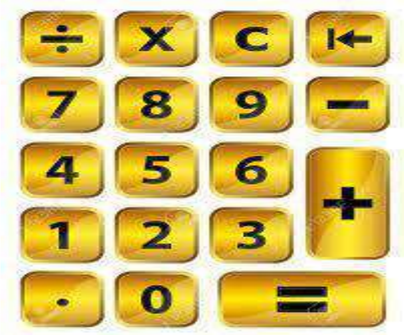
Course Learning Objectives:

1) To understand and define classes, use packages to write object oriented programming.
2) To apply the concepts of overloading and overriding in object oriented programming.
3) To apply the concepts of access specifiers and exception handling.
4) To develop applications using generic programming and event handling.
5) To build software development skills using java programming for real-world applications.

Laboratory Programs:

PART- A

1.	Develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff. If the type of the EB connection is domestic, calculate the amount to be paid as follows: First 100 units - Rs. 1 per unit 101-200 units - Rs. 2.50 per unit 201 -500 units - Rs. 4 per unit > 501 units - Rs. 6 per unit If the type of the EB connection is commercial, calculate the amount to be paid as follows: First 100 units - Rs. 2 per unit 101-200 units - Rs. 4.50 per unit 201 -500 units - Rs. 6 per unit > 501 units - Rs. 7 per unit
2.	Develop a java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary
3.	Write a program to demonstrate run-time polymorphism of overriding by implementing Rectangle and Triangle classes extended from an abstract class Figure.
4.	Write a program to demonstrate usage of multilevel inheritance by implementing Box, BoxWeight and Shipment classes with overloaded constructors.
5.	Write a program to demonstrate all combinations of the access control modifiers.
6.	a. Write a program to catch IllegalAccessException thrown inside a called method. b. Write a program to demonstrate finally block in case of i) No exception ii) Exception iii) return statement
7.	Implement a Java based program to handle all the mouse based events with appropriate display.
8.	Implement a calculator using event-driven programming paradigm of Java having the following operation:

	 ÷ x C I← 7 8 9 − 4 5 6 + 1 2 3 . 0 =
9.	Write a java program that implements a multi-threaded application that has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
10.	Write a Java Program to implement producer consumer problem using interthread communication.
	PART B – Problem Based Learning
Case Study: Develop any GUI based application using Java concepts.	

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Java: The Complete Reference	Herbert Schildt	TATA McGraw-Hill Publications	7th Edition, 2009
Reference Books				
1	The Java^{TM} Programming Language	Ken Arnold, James Gosling, David Holmes	Addison Wesley	4th Edition, 2005
2	Java Puzzlers: Traps, Pitfalls, and Corner Cases	Joshua Bloch, Neal Grafter	Pearson India Education Services Pvt. Ltd.	1st Edition 2005

Web links/Video Lectures/MOOCs/papers

1. https://www.youtube.com/watch?v=-HafzawNlUo
2. https://www.youtube.com/watch?v=7GwptabrYyk
3. https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in- java/

Course Outcomes: At the end of the course the student will be able to:	
22CDS36L.1	Understand and make use of classes, strings and basic data types in writing object oriented programming.
22CDS36L.2	Implement java programs with constructors and method overloading concepts.
22CDS36L.3	Implement applications using inheritance and method overriding concepts.
22CDS36L.4	Implement applications using packages and interfaces enforcing access controls.
22CDS36L.5	Implement programs using multithreading and exception handling constructs.
22CDS36L.6	Design and develop simple java applications for real world problems.

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CDS36L.1	3		1											
22CDS36L.2			2										2	
22CDS36L.3	3					2								
22CDS36L.4	3					1							1	
22CDS36L.5	3	2	3											
22CDS36L.6	1	2	3	3	3	3					2	2		

1: Low 2: Medium 3: High

Universal Human Values- II

Course Code	**22UHV37**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	**02**	Exam Hours	02

Course Learning Objectives: This introductory course input is intended:

1. To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. 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 Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement toward value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

Module-1 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 Fulfill the Basic Human Aspirations.

Activities: Sharing about Oneself, Exploring Human Consciousness and Exploring Natural Acceptance. **5 Hours**

Module-2 – Harmony in the Human Being

Understanding Human beings 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.

Activities: Exploring Sources of Imagination in the Self, Exploring Harmony of Self with the Body and Exploring the difference of Needs of Self and Body. **5 hours**

Module 3 – Harmony in the Family and Society

Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.

Activities: Exploring the Feeling of Trust, Exploring the Feeling of Respect and Exploring the Feeling systems to fulfil Human Goal. **5 hours**

Module-4 – 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.

Activities: Exploring the Four Orders of Nature and Co-existence in Existence. **5 hours**

Module-5 – 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

Activities: Exploring Ethical Human Conduct**,** Humanistic Models in Education and steps of Transition towards Universal Human Order. **5 hours**

Course Outcomes: At the end of the course the student will be able to:	
22UHV37.1	Practice the method of self-exploration to understand the basic human aspiration.
22UHV37.2	Distinguish between needs of self and body.
22UHV37.3	Evolve a program for self-regulation and health.
22UHV37.4	Differentiate between the characteristics and activities of different orders and study the mutual fulfillment among them.
22UHV37.5	Realize sustainable solutions to the problems in society and nature.
22UHV37.6	Develop competence in professional ethics and strategies for the transition towards a value-based life/profession.

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Foundation Course in Human Values and Professional Ethics	R R Gaur, R Asthana, G P Bagaria	Excel Books, New Delhi	2nd Revised Edition, 2019
2	Teachers' Manual for A Foundation Course in Human Values and Professional Ethics	R R Gaur, R Asthana, G P Bagaria	Excel Books New Delhi	2nd Revised Edition, 2019
Reference Books				
1	Jeevan Vidya: Ek Parichaya	A Nagaraj	Jeevan Vidya Prakashan Amarkantak	1999
2	Human Values	A.N. Tripathi	New Age Intl. Publishers, New Delhi	2004

Additional Resources/Web links/Video Lectures

1. The Story of Stuff (Book).
2. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
3. Small is Beautiful - E. F Schumacher.
4. Slow is Beautiful - Cecile Andrews
4. Economy of Permanence - J C Kumarappa
5. Bharat Mein Angreji Raj – Pandit Sunderlal
6. Rediscovering India - by Dharampal
7. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
8. India Wins Freedom - Maulana Abdul Kalam Azad
9. Vivekananda - Romain Rolland (English)
10. Gandhi - Romain Rolland (English)

11. UHV-I Teaching material (Presentations, Pre & Post Surveys etc.)
https://fdp-si.aicte-india.org/AicteSipUHV_download.php

12. Details of UHV-II: Universal Human Values – Understanding Harmony and Ethical Human Conduct
https://drive.google.com/file/d/1cznDaqDwKy__EKWmqJLWF94MeY4AXcsU/view?usp=sharing

13. Recorded FDP (Refresher 1 Part 1: Preparing to teach UHV-I in SIP)
https://www.youtube.com/watch?v=kejuD4faDDE&list=PLWDeKF97v9SOjS4RanhaYj4YLiImqm5pj&index=1

14. Resources, including the class notes and presentations
https://drive.google.com/drive/folders/1nh9m5ibEtvMyqekeiexAJtfbdNtmtt6-?usp=sharing

15. Hindi Recording of 5-day UHV FDP
https://www.youtube.com/playlist?list=PLWDeKF97v9SMRfe5PK1HPYnEcrrJOL6K7

16. English Recording of 5-day UHV FDP
https://www.youtube.com/playlist?list=PLWDeKF97v9SP7wSlapZcQRrT7OH0ZlGC4

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22UHV37.1						3		2						
22UHV37.2						2			3					
22UHV37.3						2		3						
22UHV37.4							3							
22UHV37.5			3				2							
22UHV37.6								3				2		

1: Low 2: Medium 3: High

Biology for Engineers

Course Code	**22BFE37**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	**02**	Exam Hours	02

Course Learning Objectives:

1. To bring awareness of biological concepts to engineering students
2. To introduce the building blocks of life and their complexity
3. To encourage interdisciplinary studies and projects
4. To appreciate the discoveries that mimic nature and its working
5. To inculcate nature-inspired design and operational principles

Module-1 **5 Hours**

Basic Cell Biology: Introduction to Biology, The cell: the basic unit of life, Expression of genetic information-protein structure and function, Cell metabolism; Cells respond to their external environments, Cells grow and reproduce, Cellular differentiation.

Module-2 **5 Hours**

Biochemistry and Molecular Aspects of Life: Biodiversity-Chemical bonds in Biochemistry; Biochemistry and Human biology, Protein synthesis -DNA; RNA, Transcription and translation factors play key roles in protein synthesis, Differences between eukaryotic and prokaryotic protein synthesis, Stem cells and their applications.

Module-3 **5 Hours**

Bioinspired Engineering based on human physiology: Circulatory system (artificial heart, pacemaker, stents), Nervous system (Artificial neural network), Respiratory system, sensory system (electronic nose, electronic tongue), Visual and auditory prosthesis (Bionic eye and cochlear implant).

Module-4 **5 Hours**

Relevance of Biology as an interdisciplinary approach: Biological observation that led to major discoveries, Echolocation (ultrasonography, sonars), Photosynthesis (photovoltaic cells, bionic leaf), Bird flying (aircraft), Lotus leaf effect (Super hydrophobic and self-cleaning surfaces), Plant burrs (Velcro).

Module-5 **5 Hours**

Bioinspired Algorithms and Applications: Genetic algorithm, Gene expression modelling, Parallel Genetic Programming: Methodology, History, and Application to Real-Life Problems, Dynamic Updating DNA Computing Algorithms, Bee-Hive: New Ideas for Developing Routing Algorithms Inspired by Honey Bee Behaviour.

Course Outcomes: At the end of the course the student will be able to:

22BFE37.1	Discuss how the cell forms the basic building block of life
22BFE37.2	Distinguish between transcription and translation
22BFE37.3	Describe the role played by proteins within the cell
22BFE37.4	Analyze the role of bioinspired design in novel applications
22BFE37.5	Apply bioinspired design principles to other domains
22BFE37.6	Implement a simple genetic algorithm

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Text Books				
1	Biology for Engineers	Thyagarajan.S., Selvamurugan. N., Rajesh.MP, Nazeer RA, Richard W. Thilagaraj, Barathi.S., and Jaganthan.M.K	Tata McGraw Hill	2012
2	Molecular Biology	Robert Weaver	McGraw-Hill	5th Edition, 2012
Reference books				
1	Lewin's Genes XII	Jocelyn E. Krebs, Elliott S. Goldstein, Stephen T. Kilpatrick	Jones and Bartlett Learning	2017
2	Bioinspired Engineering	Jenkins, C.H.	Momentum Press	2012
3	Bio mimetics: Nature-Based Innovation	Yoseph Bar-Cohen	CRC Press	1st Edition, 2016
4	A Practical Guide to Bio-inspired Design	Hashemi Farzaneh, Helena, Lindemann, Udo	Springer	2019

Web links/Video Lectures/MOOCs

1. https://books.google.co.in/books?id=-2LNBQAAQBAJ&printsec=frontcover#v=onepage&q&f=false

2. https://www.aminotes.com/2017/02/biology-for-engineers-module-1-cocepts.html

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22BFE37.1	2					1								
22BFE37.2		1				1								
22BFE37.3	2					2								
22BFE37.4		2										2		
22BFE37.5	2											2		
22BFE37.6		2										2		

1: Low 2: Medium 3: High

IOT ENABLED PROTOTYPING			
Course Code:	**22IEP38**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	**01**	Exam Hours	02

Course Learning Objectives:

1. Understand the IoT concepts such as sensing, actuation, and communication.
2. Development of Internet of Things (IoT) prototypes—including devices for sensing, actuation, processing, and communication and Protocols
3. Understand the significance of Project Management and the different techniques of planning
4. To introduce fundamental aspects of intellectual property rights, Govt. policies on IPR, and patentability search techniques.

Module 1

Internet of Things – Hardware / System Design

Introduction to IoT fundamentals, Introduction to sensors, Difference between analog and Digital sensors, Interfacing Temperature, Light and Humidity sensor with Arduino, Interfacing Motors with Arduino, A simple program to control actuator based on the analog sensor. **6 Hours**

Module 2

Internet of Things

Networking in IoT:

Introduction to wireless communication, Wifi Module ESP8266 interface with Arduino, Machine to Machine (M2M) communication using WiFi module. A simple demonstration of sensing temperature from one device and control actuator on a second device (M2M)

IoT in Web/ Cloud Platform:

Introduction to a web server - XAMPP(windows), A simple interactive web page using HTML5, Bootstrap (or CSS), and Javascript. Interfacing ESP8266 with webserver, ThingSpeak API, and MQTT protocol, A simple project to demonstrate the status of two IoT devices communicating with a Web Server. **6 Hours**

Module 3

Project Planning and Management

Project initiation, Project charter, Project planning, and implementation, Scheduling and costing, Project monitoring and control, Project closure and reports.

6 Hours

Module 4

Intellectual Property Rights

Introduction and the need for intellectual property right (IPR) – Kinds of Intellectual Property Rights, Elements of Patentability: Novelty, Non-Obviousness (Inventive Steps), Industrial Application, Non - Patentable Subject Matter, Registration Procedure, Patentability search methods, Patent landscape, Freedom-to-market, National IPR Policy, Govt. initiatives and scheme in promoting IPR. **6 Hours**

Course Project

Develop IoT-based prototypes (solutions) to solve any industrial or societal problems. The prototype building is teamwork of 3-5 students. The goals should be clearly defined and should use robust technologies and rigorous testing. **6 Hours**

Course Outcomes: At the end of the course, the student will be able to:

22IEP38.1	Analyze the basics of IoT and protocols.
22IEP38.2	Develop IoT-based prototypes to solve industrial and societal problems.
22IEP38.3	Apply appropriate approaches to plan a new project and develop a project schedule.

22IEP38.4	Discuss the ethical aspects in IPR, Govt. policies on IPR, and conducting patentability searches.
22IEP38.5	Inculcate the teamwork and communication skills.

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Reference Books				
1	Internet of Things (A Hands-on-Approach)	Vijay Madisetti and Arshdeep Bahga	Orient Blackswan Private Limited	1st Edition, 2015
2	Fundamentals of Intellectual Property	Dr. Kalyan C. Kankanala	Asia Law House	1st Edition, 2012
3	Project Management Absolute Beginner's Guide	Greg Horine	Pearson Education (US)	4th Edition, 2017

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22IEP38.1			**2**		**2**				**2**	**2**				
22IEP38.2			**2**								**3**			
22IEP38.3					**2**						**2**			
22IEP38.4								**1**		**2**				
22IEP38.5								**1**	**2**	**2**				

1: Low 2: Medium 3: High

Industry Oriented Training - Business Etiquettes

Course Code	**22ITB39A**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02

Course Learning Objectives:

- Know the components of self-introduction
- Develop a resume with the inclusion of core competencies
- Involve and contribute to group discussions
- Develop effective communication to succeed in the professional career
- Know the etiquettes of digital communication

Module-1

Self-Introduction &Essentials of grooming

Self-Introduction: Learn the secret to introducing Yourself, Things to avoid when introducing yourself. Activity: Video record the self-introduction. **Essentials of grooming:** Creating the first impression, what does the well-dressed man wear? What does the well-dressed woman wear? Personal hygiene and habits. **4 Hours**

Module-2

Resume Writing

Purpose, Identifying Relevant Competencies, Understanding Applicant Tracking Systems, Lists of Competencies, Writing Accomplishment/ Objective Statements, Finding the Right Words- Action verbs, The Most Popular Resume Format, Other Popular Resume Formats, Do's and Don'ts. Activity: Students have to submit a copy of their resume. **4 Hours**

Module-3

Group Discussion

Types, process, Evaluation criteria, Do's and Don'ts Activity: Group discussions have to be held during the training sessions. **4 Hours**

Module-4

Communicate effectively

Build a Story, Just a Minute, Group Activities, Team building activities, Role Play, Presentation Skills. **4 Hours**

Module-5

Digital right and wrong

Virtual Communication: Agenda, being prepared, Dressing appropriately, background, Use Microphone and camera the right way, restraining from off tasks during virtual meetings, protecting confidential data during online presentations, time management. **4 Hours**

Course Outcomes: At the end of the course the student will be able to:

22ITB39A.1	Articulate the essential components required for self-introduction in any business or a networking event and also recognize the need to dress appropriately for a successful career in the corporate
22ITB39A.2	Develop a resume inclusive of core competencies, and action verbs which are compatible with Applicant Tracking Systems
22ITB39A.3	Demonstrate the types, process and evaluation process of Group Discussion and carry out effective group discussions
22ITB39A.4	Develop skills required for effective communication
22ITB39A.5	Associate and be accustomed to the etiquette to be followed during online meetings

Sources

1. English for Common Interactions in the Workplace: Basic Level: Coursera: https://www.coursera.org/learn/english-common-interactions-workplace-basic-level
2. Personal Communication-Introduce Yourself With Confidence: https://www.udemy.com/course/how-to-introduce-yourself/
3. Professionalism, Grooming and Etiquette: https://www.edx.org/course/professionalism-grooming-and-etiquette
4. How to Write a Resume: https://www.coursera.org/learn/how-to-write-a-resume#syllabus
5. Group Discussion Strategies: https://www.udemy.com/course/group-discussion-strategies/
6. Communication Strategies for a Virtual Age: https://www.coursera.org/learn/communication-strategies-virtual-age#syllabus

References

1. https://simplifytraining.com/course/personal-hygiene-and-good-grooming/
2. https://www.udemy.com/course/group-discussion-strategies/
3. https://www.educba.com/course/group-discussion/
4. https://getrafiki.ai/meetings/rules-of-virtual-meeting-etiquette-every-sales-professional-should-follow/
5. https://thedigitalworkplace.com/articles/online-meeting-etiquette-for-attendees/
6. https://rigorousthemes.com/blog/virtual-meeting-etiquette-guidelines-ground-rules/

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22ITB39A.1									2	3		1		
22ITB39A.2										3		1		
22ITB39A.3									2	3	1	1		
22ITB39A.4									2	3	1	1		
22ITB39A.5									2	3	1	1		

1: Low 2: Medium 3: High

Industry Oriented Training - Computing Skills			
Course Code	**22ITC39B**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02

Course Learning Objectives:

1. Use logical conditions for problem-solving and also introduce the concepts of arrays
2. Know functions, function calls, and parameter passing
3. Introduce algorithms and appreciate their importance in problem-solving
4. Introduce the core concepts of OOP's
5. Differentiate between front-end & back-end development and recognize the use of database management

Module-1

Introduction to computing constructs

Logical conditions: For Loops, Nested For Loops, While Loops, Do-While Loops, Nesting and Boxes, and combine/negate several logical conditions using logic operations AND, OR, and NOT.

Arrays & strings: Create arrays of characters (strings), use the null terminator, and manipulate strings. **4 Hours**

Module-2

Functions & Pointers

Introduction to Functions, Returning Data From a Function, Passing Data Into a Function, Getting Valid User Input, Changing Parameter Values, Pointer Basics, Changing the Pointed to Value, Walking an Array with Pointers, Dynamic Memory Allocation, Getting More Memory, Pointers to Structure. **4 Hours**

Module-3

Algorithm analysis

Introduction to Algorithm Analysis, Big-O, Big-O Examples, Dynamic Array Operations, Bubble Sort, Selection Sort, Insertion Sort, Recursion, Recursive Binary Search, Merge Sort.

4 Hours

Module-4

Object-oriented programming

Designing for Object-Oriented Programming, Core Concepts of OO Programming: Classes and objects, data abstraction, encapsulation, inheritance, benefits of inheritance, polymorphism, procedural and object-oriented programming paradigm. **4 Hours**

Module-5

Frontend and backend development

UI, Database management: DBMS overview, Relational Data Model and the CREATE TABLE Statement, Basic Query Formulation with SQL. **4 Hours**

Course Outcomes: At the end of the course the student will be able to:	
22ITC39B.1	Illustrate the use of logical conditions, declare and manipulate data into arrays
22ITC39B.2	Implement functions, function calls, and parameter passing
22ITC39B.3	Design, implement, and evaluate an algorithm to meet desired needs
22ITC39B.4	Describe the core concepts of OOP's
22ITC39B.5	Recognize the concepts of front-end development and database management

Sources

1. Computational Thinking with Beginning C Programming Specialization: https://www.coursera.org/learn/simulation-algorithm-analysis-pointers?specialization=computational-thinking-c-programming#syllabus
2. Simulation, Algorithm Analysis, and Pointers: https://www.coursera.org/lecture/simulation-algorithm-analysis-pointers/big-o-examples-pdCan
3. Programming Fundamentals: https://www.coursera.org/learn/programming-fundamentals?specialization=c-programming#syllabus
4. Object-Oriented Programming Concepts: https://www.coursera.org/learn/concepts-of-object-oriented-programming#syllabus
5. Introduction to Back-End Development: https://www.coursera.org/learn/introduction-to-back-end-development

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22ITC39B.1	2	1	1											
22ITC39B.2	2	1	1											
22ITC39B.3	1	1	2											
22ITC39B.4	2		1											
22ITC39B.5	2	1	1											

1: Low 2: Medium 3: High

IV Semester

Linear Algebra and Statistical Methods

Course Code	22CDS41	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	2:2:0	SEE	3 Hours
Total Hours	40	Credits	03

Course Learning Objectives:

1. To apply principles of linear algebra in Engineering applications.
2. To use probability theory and random process for applications in Computer Science and related domains.

Module-1 **8 Hours**

Linear mappings and matrices:

Linear mappings and linear transformations, matrix representation of a linear operator, Change of basis, similarity of matrices. Matrix representations of general linear mappings.

Module-2 **8 Hours**

Inner product spaces, orthogonality:

Inner product, length & orthogonality, orthogonal set, orthogonal projection Gram-Schmidt process, QR factorization of matrices. Diagonalization of symmetric matrices. The singular value decomposition.

Module-3 **8 Hours**

Statistical Methods and Curve Fitting:

Correlation and Regression-Karl Pearson's coefficient of correlation-problems. Regression analysis- lines of regression -problems and Rank Correlation-problems.

Curve Fitting: Curve fitting by the method of least squares-fitting the curves of the form $y = ax + b$, $y = ax^2 + bx + c$ and $y = ax^b$

Module-4 **8 Hours**

Probability Distributions:

Random variables (discrete and continuous), probability mass/density functions, cumulative density function. Binomial, Poisson, exponential and normal distributions- problems (No derivation for mean and standard deviation)

Module-5 **8 Hours**

Sampling theory:

Introduction, sampling distributions, Testing of hypothesis for means, level of significance, confidence limits, Sampling of variables, central limit theorem, confidence limits for unknown mean, student’s t-distribution, Chi-square distribution as a test of goodness of fit.

Course Outcomes: At the end of the course the student will be able to:

22CDS41.1	Apply Linear transformation technique in machine learning algorithms. Feature scaling and normalization. Additionally, principal component analysis (PCA).
22CDS41.2	Apply the technique of singular value decomposition for data compression and least-square approximation in solving inconsistent linear systems.
22CDS41.3	Make use of the correlation and regression analysis to fit a suitable mathematical model for the statistical data.
22CDS41.4	Discover the relation between dependent & independent variables using the least square curve fitting method.
22CDS41.5	Develop probability distribution of discrete, continuous random variables occurring in engineering domains.

22CDS41.6	Demonstrate the validity of testing the hypothesis to arrive at a decision regarding the population through a sample.

Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Schaum's Outline of Linear Algebra	Seymour Lipschutz and Marc Lipson	McGraw Hill Education	3 rd Edition, 2022
2	Linear Algebra & its applications	David C. Lay	Pearson Publication	3^{rd} Edition, 2014
3	Higher Engineering Mathematics	B.S. Grewal	Khanna Publishers	44^{th} Edition, 2017.
Reference Books				
1	Advanced Engineering Mathematics	C.Ray Wylie, Louis C.Barrett	McGraw- Hill Book Co., New York	6^{th} Edition,2017
2	Probability & Statistics for Engineers & Scientists	Ronald Walpole, R H Myers, S L Myers and K. Ye	Pearson Publication	9^{th} Edition, 2016
3	Linear Algebra & its applications	Gilbert Strang	Cengage Learning India Edition	4^{th} Edition, 2006

Web links and Video Lectures (e-Resources):

- https://www.youtube.com/watch?v=LJLoJhbBA4&list=PLbMVogVj5nJQ2vsW_hmyvVfO4GYWaaPp7
- https://www.youtube.com/watch?v=1Q7x7UmlORs
- https://www.youtube.com/watch?v=n6rD54jIzOU
- https://www.youtube.com/watch?v=V8F8We_-nuo&list=PLhSp9OSVmeyLB62_-fT9VNbjRkDEzJzzp
- https://www.youtube.com/watch?v=-WfQ-T6Sd7s

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS41.1	1	2												
22CDS41.2	1	2												
22CDS41.3	2	1												
22CDS41.4		2										1		
22CDS41.5	2	1												
22CDS41.6	2	1												

1: Low 2: Medium 3: High

Design and Analysis of Algorithms

Course Code	**22CDS42**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Integrated	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	2:2:2	SEE	3 Hours
Total Hours	40 hours Theory + 10 Lab slots	Credits	04

Course Learning Objectives: The objective of the course is to

1. Apply appropriate methods to solve a given problem and validate its correctness solving.
2. Analyze time complexity of the algorithms.
3. Implement various algorithmic techniques like Greedy strategy, Divide and Conquer approach, Dynamic Programming and Backtracking.
4. Understand Synthesizing efficient algorithms in common engineering design situations.
5. Know the limitations of algorithmic power.

Module-1: Introduction & Analysis Framework **8 hours**

Introduction: What is an Algorithm? Fundamentals of Algorithmic Problem Solving, Important Problem Types. **Analysis:** Analysis Framework, Asymptotic Notations and Basic Efficiency classes, Mathematical analysis of Non-Recursive and Recursive Algorithms with Examples. Empirical Analysis of Algorithms**. TB1**: **Ch 1.1,1.2,1.3,2.1 to 2.6**

Module-2: Divide and Conquer **8 hours**

Divide and Conquer: General method, Recurrence equation, Master Theorem, Merge sort, Quick sort, Strassen's matrix multiplication. **Decrease and Conquer:** Binary search. **Transform and Conquer:** AVL Trees, Heaps and Heap sort. **TB1**: **Ch 5.1,5.2,5.4,4.4,6.3,6.4**

Module-3: Greedy Method **8 hours**

Greedy method: General method. **Minimum cost spanning trees:** Prim's Algorithm, Kruskal's Algorithm. **Single source shortest paths:** Dijkstra's Algorithm. **Optimal Tree Problem:** Huffman Trees and Codes. **Space and Time Tradeoffs:** Sorting by Counting, B- Trees. **TB1: Ch 9, 7.1,7.4**

Module-4: Dynamic programming **8 hours**

Dynamic programming: Knapsack problem with memory functions, Optimal Binary Search Trees, Transitive Closure-Warshall's Algorithm, All Pairs Shortest Paths-Floyd's Algorithm.
Limitations of Algorithm Power: P, NP and NP- Complete Problems. **TB1: Ch 8.2,8.3, 8.4,11.3**

Module-5: Backtracking, Branch and Bound **8 hours**

Backtracking: N-Queens problem, Hamiltonian circuit Problem, Sum of subsets problem.
Branch and Bound: Assignment problem, Knapsack problem, Travelling Sales Person problem.
TB1: Ch 12.1,12.2

PRACTICAL MODULE

Implement the specified algorithms for the following problems using Java. IDE's such as NetBeans / Eclipse can be used for development and demonstration.

- Binary Search: To search a key in the list of n integers.
- Merge Sort: To sort n randomly generated integers.
- Quick Sort: To sort n randomly generated integers.
- Prim's algorithm: To find the Minimum Spanning Tree of an undirected graph.
- Kruskal's Algorithm: To find the Minimum Spanning Tree of an undirected graph.
- Floyd's Algorithm: To find all pairs shortest distance in a graph.
- Knapsack Problem: To solve 0/1 Knapsack problem using dynamic programming
- Subset problem: To solve the sum of subset problem using branch and bound method
- Open ended experiment covering the concept of entire syllabus: Online shopping application

Course Outcomes: At the end of the course the student will be able to:	
22CDS42.1	Interpret the time and space complexity of algorithms which provides solutions to the given problem
22CDS42.2	Identify the problems from the set that can be solved using divide and conquer techniques and apply the technique to obtain the solutions.
22CDS42.3	Apply the technique of greedy algorithms in real life applications to get the optimal solution
22CDS42.4	Apply the dynamic programming design technique to solve various problems
22CDS42.5	Differentiate the problems that can be solved using backtracking method and other general design techniques for given set of problems
22CDS42.6	Analyze the limitations of algorithm power

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Introduction to the Design and Analysis of Algorithms	Anany Levitin	Pearson	3rd Edition, 2012
Reference Books				
1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein	PHI Learning	3rd Edition, 2010
2	Computer Algorithms	Ellis Horowitz, Satraj Sahni and Rajasekaran	PHI Learning	2nd Edition, 2019

Web links and Video Lectures (e-Resources):

1. NPTEL Design and Analysis of Algorithms by Prof. Madhavan Mukund, https://nptel.ac.in/courses/106106131
2. NPTEL Fundamental Algorithms: Design and Analysis by Prof. Sourav Mukhopadhyay, https://onlinecourses.nptel.ac.in/noc22_cs01/preview
3. GeekforGeeks, Algorithms https://www.geeksforgeeks.org/fundamentals-of-algorithms/
4. Turorialspoint, Design and Analysis of Algorithms Tutorial https://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm
5. https://www.youtube.com/watch?v=0IAPZzGSbME&list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS42.1		**2**		**3**										**3**
22CDS42.2		**3**	**3**										**2**	
22CDS42.3			**3**										**2**	
22CDS42.4			**3**										**2**	
22CDS42.5				**3**									**2**	
22CDS42.6				**3**										**3**

1: Low 2: Medium 3: High

Database Management System (Integrated)

Course Code	**22CDS43**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	**04**	Exam Hours	03

Course Learning Objectives:

- Provide a strong foundation in database concepts, technology, and practice.
- Practice SQL programming through a variety of database problems.
- Demonstrate the use of concurrency and transactions in the database.
- Design and build database applications for real world problems.
- Develop applications to interact with databases.

Module-1: Database & its Concepts — 8 Hours

Introduction to Databases: Introduction, Simplified database system environment, Characteristics of database approach, Actors on the scene, Workers behind the scene, Advantages of using the DBMS approach.

Database Concepts and Architectures: Data Models, Schemas and Instances. Three schema architecture and data independence, database languages and interfaces, Component modules of a DBMS and their Interactions.

Data Model: Main phases of a Database Design Process, Entity Types, Entity Sets, Attributes, Keys, Relationship Types, Sets, Roles and Structural Constraints, ER diagram Notations and examples.

(Chapter No: 1.1, 1.3, 1.4, 1.5, 1.6, 2.1, 2.2, 2.3, 2.4.1, 3.1, 3.3.1, 3.3.2, 3.4, 3.7)

Module-2: Relational Model & SQL — 8 Hours

Relational Model: Relational Model Concepts, Relational Model Constraints and schemas, Update Operations and Dealing with Constraint violations.

SQL: SQL data definition and data types, Specifying Constraints in SQL, Basic Retrieval Queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, More Complex SQL Retrieval Queries, Specifying Constraints as Assertions and Triggers, Views in SQL, Schema Change Statements in SQL.

(Chapter No: 5.1, 5.2, 5.3, 6.1, 6.2, 6.3, 6.4, 7.1, 7.2, 7.3, 7.4)

Module-3: Relational Algebra & Design — 8 Hours

Relational Algebra & Design: Unary and Binary relational operations, Relational Algebra Operations, Additional Relational Operations, Examples of Queries in Relational Algebra.

Database Application Development: Embedded SQL, Dynamic SQL, SQLJ, Database Programming with Function calls: SQL and JDBC, Database Stored Procedures.

(Chapter No. 8.1 to 8.5, 10.2, 10.3, 10.4)

Module-4: Normalizations — 8 Hours

Normalization and its Algorithms: Informal design guidelines for relation schema, Functional Dependency (Inference Rules, Equivalence, and Minimal Cover) , Normal Forms based on Primary Keys, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form, Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, Dangling tuples.

(Chapter No: 14.1 to 14.7, 15.2 to 15.4)

Module-5: Transaction Processing, Database Recovery&Security — 8 Hours

Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Two-phase locking techniques for Concurrency control.

Database Recovery Protocols: Recovery Concepts, NO-UNDO/REDO algorithm, Recovery techniques based on immediate update, Shadow paging, ARIES recovery algorithm.

Database Security: Database security issues-Types of security and control measures, SQL Injection, Challenges to Maintaining Database Security.
(Chapter No: 20.1 to 20.5, 21.1, 22.1 to 22.5, 30.1, 30.4, 30.9)

List of Laboratory Experiments related to above modules – 2 hours each

1. Demonstrating Database Connectivity in java, php, python (using anyone of these) .

- Design a Database for e.g. Bank Database, College Database. Mention the actors on the scene and workers behind the scenes for these two database applications. Write the scheme diagram of these databases. Design a suitable interface for each category of users. (Drawing tool like "**drawio**" can be explored if required)
- Write an ER diagram for e.g. Bank Database, College Database. Create required tables. Apply the constraints like Primary Key, Foreign key, NOT NULL to the tables. Insert sample data into each table.
- Demonstrate mapping of ER-diagram to Relational schema model. (Chapter No. 9.1)
- Perform the operation for demonstrating the insertion, updation and deletion using the referential integrity constraints. Write the query to create the view, Altering the existing view and dropping view.
- Write a SQL statement for implementing ALTER, UPDATE and DELETE. Write the query for implementing the aggregate functions like: MAX(), MIN(), AVG(), COUNT(), SUM().
- Perform queries involving predicates LIKE, BETWEEN, IN etc. Write the queries to implement different types of joins.
- Write queries to solve the concept of nested query, correlated query and Group by clauses.
- Write a suitable query for Assertion, Triggers and Cursor.
- Study transaction control commands like Commit, Rollback, Save point, Set Transaction and perform its execution. Write the query for creating the users and their role.

Course Outcomes: At the end of the course the student will be able to:	
22CDS43.1	Identify the various elements of Database Management Systems and to draw an E-R diagram.
22CDS43.2	Solve a given problem statement, analyze the entities, its types and their relations.
22CDS43.3	Take part in writing Queries using Relational Algebra, SQL and PL/SQL.
22CDS43.4	Examine the normalizations for the development of application software.
22CDS43.5	Determine the concepts of transaction, concurrency control, recovery and security in the database.
22CDS43.6	Develop a database application system using advanced SQL tools and interfaces with appropriate documentation.

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Fundamentals of Database Systems	Ramez Elmasri and Shamkant B. Navathe,	Pearson	7th Edition, 2017

Reference Books				
1	Database System Concepts	Abraham Silberschatz, Henry F. Korth, S. Sudharshan	Tata McGraw Hill	6th Edition, 2011
2	Database Management Systems	Ramakrishnan, and Gehrke	Tata McGraw Hill	3rd Edition, 2014
3	Modern Database Management	Hoffer, Ramesh, Topi	Pearson	12th Edition, 2016

Web links/Video Lectures/MOOCs/papers

1. https://www.tutorialspoint.com/dbms/
2. https://www.w3schools.com/sql/
3. https://www.codecademy.com/learn/learn-sql
4. https://in.udacity.com/
5. https://www.geeksforgeeks.org/dbms/

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS43.1	2	2	-	-	1	-	-	-	-	-	-	-	-	1
22CDS43.2	2	-	-	-	-	2	-	-	-	-	-	-	-	-
22CDS43.3	-	2	2	-	-	-	-	-	-	-	-	-	-	1
22CDS43.4	-	-	-	-	-	1	-	-	-	-	-	-	-	1
22CDS43.5	-	-	2	-	-	-	1	-	-	-	-	-	1	1
22CDS43.6	-	-	-	-	-	1	-	-	1	-	-	-	1	-

1: Low 2: Medium 3: High

Operating System

Course Code	**22CDS44**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	3:0:0	SEE	3 Hours
Total Hours	40 Hours	Credits	03

Course Learning Objectives: The objective of the course is to

- Demonstrate need for Operating system, types and services.
- Apply suitable techniques for management of various resources.
- Use Processor, Memory, Storage and File system commands.
- Demonstrates the use of Memory and Virtual memory management.
- Analyze the concept of Deadlock and Process synchronization.

Module-1: **8 hours**

Introduction to operating systems, System structures: What Operating Systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. Operating System Services: User – Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.

TB1:**Ch**1, 2

Module-2: **8 hours**

Process Management Process concept; Process scheduling; Operations on processes; Inter process communication Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. CPU Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling.

TB1: Ch 3, 4, 6

Module-3: **8 hours**

Process Synchronization: Synchronization: The critical section problem; Peterson’s solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors. Deadlocks: Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

TB1: Ch 5,7

Module-4: **8 hours**

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

TB1: Ch 8,9

Module-5: **8 hours**

File System, Implementation of File System: File system: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.

Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems.

TB1: Ch 10, 12, 13,14

Course Outcomes: At the end of the course the student will be able to:	
22CDS44.1	Analyze the design of Operating System operations and Services.
22CDS44.2	Demonstrates Process Management and Multi-threaded programming.
22CDS44.3	Illustrate the mechanism of Process Synchronization and Deadlock.
22CDS44.4	Illustrate Memory and Virtual Memory Management.
22CDS44.5	Implementation of File System and Space allocation method.
22CDS44.6	Interpret the concepts of Secondary Storage Structure and Protection.

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Operating System Concepts	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne	Wiley-India	9th 2012 Edition
Reference Books				
1	Operating Systems: Internals and Design Principles	William Stallings	Pearson	6th Edition 2012
2	Modern Operating Systems	Andrew S. Tannenbaum and Herbert Bos	Pearson	4th Edition, 2015
3	Understanding Operating System	Ann McHoes Ida M Fylnn	Cengage Learning	6th Edition 2017
4	Operating Systems: A Concept Based Approach	D.M Dhamdhere	McGraw-Hill	3rd Edition, 2013
5	An Introduction to Operating Systems: Concepts and Practice	P.C.P. Bhatt	PHI(EEE)	4th Edition, 2014

Web links and Video Lectures (e-Resources):

- https://www.geeksforgeeks.org/operating-systems
- https://www.codingninjas.com/courses/operating-system
- https://www.udacity.com/course/introduction-to-operating-systems--ud923

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CDS44.1														2
22CDS44.2		2												
22CDS44.3		2												
22CDS44.4						2								2
22CDS44.5						2								2
22CDS44.6			2											

1: Low 2: Medium 3: High

C# Programming with Dot Net

Course Code	**22CDS451**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	3:0:0	SEE	3 Hours
Total Hours	40 Hours	Credits	03

Course Learning Objectives: The objective of the course is to

- Understand .NET framework and its runtime environment
- Gain the major aspects of C# language
- Know object-oriented programming concepts implementation
- Understand working of Exceptions and Object Lifetime

Module-1: The Philosophy of .NET 8 hours

Understanding the Previous State of Affairs, The.NET Solution, The Building Block of the .NET Platform (CLR,CTS, and CLS), The Role of the .NET Base Class Libraries, What C# Brings to the Table, An Overview of .NET Binaries (aka Assemblies), the Role of the Common Intermediate Language, The Role of .NET Type Metadata, The Role of the assembly Manifast, Compiling CIL to Platform – Specific Instructions, Understanding the Common Type System, Intrinsic CTS Data Types, Understanding the Common Languages Specification, Understanding the Common Language Runtime A tour of the .NET Namespaces, Increasing Your Namespace Nomenclature, Deploying the .NET Runtime.

TB1 : Ch-1

Module-2: Building C# Applications 8 hours

The Role of the Command Line Compiler(csc.exe), Building C# Application using csc.exe ,Working with csc.exe Response Files, Generating Bug Reports, Remaining g C# Compiler Options, The Command Line Debugger (cordbg.exe) Using the, Visual studio .NET IDE, Other Key Aspects of the VS.NET IDE, C# “Preprocessor:” Directives, an Interesting Aside: The System. Environment Class

TB1 : Ch-2

Module-3: C# Language Fundamentals 8 hours

The Anatomy of Basic C# Class, Creating objects: Constructor Basics, The Composition of a C# Application, Default assignment and Variable Scope, The C# Member Initialisation Syntax, Basic Input and Output with the Console Class, Understanding Value Types and Reference Types, The Master Node: System, Object, The System Data Types (and C# Aliases), Converting Between Value Types and Reference Types: Boxing and Unboxing, Defining Program Constants, C# Iteration Constructs, C# Controls Flow Constructs, The Complete Set of C# Operators, Defining Custom Class Methods, Understating Static Methods, Methods Parameter Modifies, Array Manipulation in C#, String Manipulation in C#, C# Enumerations, Defining Structures in C#, Defining Custom Namespaces.

TB1 : Ch-3

Module-4: Object- Oriented Programming with C# 8 hours

Forms Defining of the C# Class, Definition the “Default Public Interface” of a Type, Recapping the Pillars of OOP, The First Pillars: C#’s Encapsulation Services, Pseudo- Encapsulation: Creating Read-Only Fields, The Second Pillar: C#’s Inheritance Supports, keeping Family Secrets: The “ Protected” Keyword, Nested Type Definitions,The Third Pillar: C #’s Polymorphic Support, Casting Between.

TB1 : Ch-4

Module-5: Exceptions and Object Lifetime:	**8 hours**
Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handling, the System. Exception Base Class, Throwing a Generic Exception, Catching Exception, CLR System – Level Exception(System. System Exception), Custom Application-Level Exception(System. System Exception), Handling Multiple Exceptions, The Family Block, the Last Chance Exception Dynamically Identifying Application – and System Level Exception Debugging System Exception Using VS. NET, Understanding Object Lifetime, the CIT of "new', The Basics of Garbage Collection,, Finalization a Type, The Finalization Process, Building an Ad Hoc Destruction Method, Garbage Collection Optimizations, The System. GC Type. **TB1 : Ch-5,6**	

Course Outcomes: At the end of the course the student will be able to:	
22CDS451.1	Identify introductory programming concepts using C#
22CDS451.2	Understand Command Line Compiler and Preprocessor Directives
22CDS451.3	Determine logical alternatives with C# decision structures utilizing iteration, class methods, fields, and string manipulation
22CDS451.4	Demonstrate knowledge of object-oriented concepts
22CDS451.5	Understand ways of exception handling within the .NET application environment.
22CDS451.6	Design and Implement Windows Applications using Windows Forms, Control Library

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Pro C# with .NET 3.0	Andrew Troelsen	HarperCollins Publishers Ltd	Special Edition 2007
Reference Books				
1	Programming in C#	E. Balagurusamy	Tata McGraw Hill	5th Reprint, 2004
2	C# and .Net Platform	Andrew Troelsen	APress	1st Edition, 2001
3	Programming C#	J. Liberty	O.Reilly	1st Edition, 2001

Web links and Video Lectures (e-Resources):

- http://www.tutorialspoint.com/csharp/index.htm
- https://www.sanfoundry.com/csharp-programming-examples
- https://learn.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2012/67ef8sbd(v=vs.110)
- https://csharp-station.com/Tutorial

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CDS451.1	1	2							2					1
22CDS451.2	1	2							2					1
22CDS451.3	1	2							2					1
22CDS451.4	1	2					1		2					1
22CDS451.5	1	2					1		2					
22CDS451.6		2					1	2	2	2				

1: Low 2: Medium 3: High

Systems Programming

Course Code	**22CDS452**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Theory	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	3:0:0	SEE	3 Hours
Total Hours	40 Hours	Credits	03

Course Learning Objectives: The objective of the course is to

- Written technical communication and effective use of concepts and terminology.
- Facility with UNIX command syntax and semantics.
- Ability to read and understand specifications, scripts and programs.
- Individual capability in problem solving using the tools presented within the class.
- Students will demonstrate a mastery of the course materials and concepts within in class discussions.

Module-1: Introduction **8 hours**

UNIX and ANSI Standards: The ANSI C Standard, The ANSI/ISO C++ Standards, Difference between ANSI C and C++, The POSIX Standards, The POSIX.1 FIPS Standard, The X/Open Standards. UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

TB1:**Ch1**,

Module-2: UNIX Files and APIs **8 hours**

File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links. UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs.

TB1: Ch 2, 3

Module-3: UNIX Processes and Process Control **8 hours**

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, UNIX Kernel Support for Processes. Process Control: Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, Race Conditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection. Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, Job Control, Shell Execution of Programs, Orphaned Process Groups.

TB2: Ch 4, 5

Module-4: Signals and Daemon Processes: Signals **8 hours**

The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers. Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model.

TB2: Ch 6

Module-5: Interprocess Communication **8 hours**

Overview of IPC Methods, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores. Shared Memory, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.

TB2: Ch 7, 8

Course Outcomes: At the end of the course the student will be able to:	
22CDS452.1	Illustrate ANSI C, C++ standards, POSIX standards, UNIX & POSIX API’s for UNIX operating system.
22CDS452.2	Classify the UNIX File, File System, UNIX Kernel support for files and different types of APIs.
22CDS452.3	Demonstrate the concept of processes and process Control.
22CDS452.4	Interpret the concepts of process relationships, signal handling mechanism, daemon characteristics, coding rules and error logging.
22CDS452.5	Relate the IPC issues and techniques in UNIX system programming.
22CDS452.6	Ability to understand and reason out the working of Unix Systems. Build an application/service over a Unix system

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Unix System Programming Using C++	Terrence Chan	Prentice Hall India	1999
2	Advanced Programming in the UNIX Environment	W.Richard Stevens, Stephen A. Rago	Pearson Education / PHI	3rd edition, 2005
Reference Books				
1	Advanced Unix Programming-	Marc J. Rochkind,	Pearson Education,	2nd Edition, 2005.
2	The Design of the UNIX Operating System	Maurice.J.Bach	Pearson Education / PHI	1st Edition, 1987
3	Unix Internals	Uresh Vahalia	Pearson Education	2001

Web links and Video Lectures (e-Resources):

- https://www.guru99.com/introduction-to-shell-scripting.html
- https://www.softwaretestinghelp.com/unix-shell-scripting-tutorial/
- https://www.youtube.com/watch?v=cQepf9fY6cE
- https://www.youtube.com/watch?v=8c1BL5b47kg

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22CDS452.1	1	1										1	1	
22CDS452.2	2	1										2	2	
22CDS452.3	2	1										2	2	
22CDS452.4	2	1										2	2	
22CDS452.5	2	1										2	2	
22CDS452.6										1	1			

1: Low 2: Medium 3: High

Application Development using Python Laboratory

Course Code	**22CDS46L**	CIE Marks	50
Course Type (Theory/Practical/Integrated)	Practical	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P)	0:0:2	SEE	3 Hours
Total Hours	10 Lab Slots	Credits	01

Course Learning Objectives: The objective of the course is to

- Learn the syntax and semantics of the Python programming language.
- Illustrate the process of structuring the data using lists, tuples and dictionaries.
- Understand the String manipulation methods and operators
- Demonstrate the use of built-in functions to read/write files.
- Interpret the concepts of Object-Oriented Programming as used in Python.

Descriptions (if any):

Installation procedure of the required software must be demonstrated.

PART- A (For Practice)

1. a) Write a python program to build menu-driven simple calculator
b) Write a program using a for loop to print factorial of a given number
c) Write a python program to check whether given number is palindrome

2. a) Write a python program to swap two user input numbers using user defined functions without a third variable
b) Write a python program to find the area of square, rectangle and circle using user defined functions. Take input from the user and print the results.
c) Write a python program to demonstrate the Tower of Hanoi using a function.

3. a) Guess the Number: Write a program that tells the player that it has come up with a secret number and will give the player six chances to guess it. The code that lets the player enter a guess and checks that guess is in a for loop that will loop at most six times.
b) Collatz Sequence: Write a function named collatz() that has one parameter named number. If number is even, then collatz() should print number // 2 and return this value.If number is odd, then collatz() should print and return 3 * number + 1. Then write a program that lets the user type in an integer and that keeps calling collatz() on that number until the function returns the value 1.

4. a) Write a python program to create a to-do list application. Operations should include:
(i) create a to-do list of five tasks
(ii) search for any task, if no match found add tasks to existing list
(iii) remove any task from the list
(iv) sort the to-do list and display the tasks
b) Write a python program to create a student gradebook using lists:
(i) Enter five subject scores for each student
(ii) Calculate the average scores
(iii) Grade them based on average scores
Note: Refer the table below for grading

Avg Marks	100-95	95-90	85-90	80-85	70-80	60-70	Below 60
Grade	A+	A	B+	B	C	D	F

5. a) Write a python program to calculate the total number of an item being brought by all the guests given in the below list.
allGuests = {'Alice': {'apples': 5, 'pretzels': 12}, 'Bob': {'ham sandwiches': 3, 'apples': 2},'Carol': {'cups': 3, 'apple pies': 1}}
Output:

<table>
<tr><td></td><td>
<b>Number of things being brought:</b>
<table>
<tr><td>Apples</td><td>7</td></tr>
<tr><td>Cups</td><td>3</td></tr>
<tr><td>Cakes</td><td>0</td></tr>
<tr><td>Ham Sandwiches</td><td>3</td></tr>
<tr><td>Apple Pies</td><td>1</td></tr>
</table>
b) Write a function named displayInventory(inventory) that would take any possible “inventory” (for example, {'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12}) and display it like the following:
Inventory:
<table>
<tr><td>arrow</td><td>12</td></tr>
<tr><td>gold coin</td><td>42</td></tr>
<tr><td>rope</td><td>1</td></tr>
<tr><td>torch</td><td>6</td></tr>
<tr><td>dagger</td><td>1</td></tr>
</table>
<b>Total number of items: 62</b>
Write a function named addToInventory(inventory, addedItems) that receives inventory (ex: {'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12}) and addedItems (ex: ['gold coin', 'dagger', 'gold coin', 'gold coin', 'ruby']) and return a dictionary that represents the updated inventory.
Inventory:
<table>
<tr><td>gold coin</td><td>45</td></tr>
<tr><td>rope</td><td>1</td></tr>
<tr><td>ruby</td><td>1</td></tr>
<tr><td>dagger</td><td>1</td></tr>
</table>
<b>Total number of items: 48</b>
</td></tr>
<tr><td>6.</td><td>Write a python program to play tic-tac-toe board game</td></tr>
<tr><td>7.</td><td>a) Write a python program that accepts a sentence and finds the number of words, digits, uppercase letters and lowercase letters.
b) Write a python program that repeatedly asks users for their phone number (10 digits) and email (lowercase and uppercase letters, numbers, a dot, an underscore, a percent sign, a plus sign, or a hyphen) until they provide valid input.</td></tr>
<tr><td>8.</td><td>a) Write a python program that takes the account’s name—for instance, email or blog from the command line arguments and copies the account’s password to the clipboard so that the user can paste it into a Password field.
b) Write a python program that will get the text from the clipboard, add a star and space to the beginning of each line, and then paste this new text to the clipboard.</td></tr>
<tr><td>9.</td><td>a) Write a python program to create 3 files file1.txt, file2.txt and file3.txt in a folder. Write the content in file1.txt as “VTU” and in file2.txt as “UNIVERSITY”. Open and merge contents of file1.txt and file2.txt and write the merged content in file3.txt.
b) Write a python program that opens all .txt files in a folder and searches for any line that matches a user-supplied regular expression. If a match is found, move those files to a different folder.</td></tr>
<tr><td>10.</td><td>a) Write a python program to create a class named Matrix to represent 2D matrices. The class should allow the following operations:
(i) Use __init__ method to initialize a matrix with a list of rows and columns.
(ii) The __str__ method should return a string representation of the matrix as a grid.
(iii) Use __add__ method to return a new matrix that represents the sum of the two matrices.</td></tr>
</table>

b) Write a python program to create a Banking system with two customer instances and various banking operations. Define two classes, namely customers and accounts. The Customer class should represent a customer and allows an individual to open accounts, deposit money, withdraw money, and check balances. The Account class should represent individual accounts and handle deposits and withdrawals.

PART B – AI Problems.

Case Study:

Generating Random Quiz Files:

Say you're a geography teacher with 35 students in your class and you want to give a pop quiz on US state capitals. Alas, your class has a few bad eggs in it, and you can't trust the students not to cheat. You'd like to randomize the order of questions so that each quiz is unique, making it impossible for anyone to crib answers from anyone else. Of course, doing this by hand would be a lengthy and boring affair and hence we need a python program to automate the task. Here is what the program does:

- Creates 35 different quizzes.
- Creates 50 multiple-choice questions for each quiz, in random order.
- Provides the correct answer and three random wrong answers for each question, in random order.
- Writes the quizzes to 35 text files.
- Writes the answer keys to 35 text files.

After you run the program, this is how your capitalsquiz1.txt file will look, though of course your questions and answer options may be different from those shown here, depending on the outcome of your random.shuffle() calls:

Name :
Date :
Period :

State Capitals Quiz (Form-1)

1. What is the capital of Virginia ?

A. Hartford

B. Santa Fe

C. Harrisburg

D. Charleston

2. What is the capital of Colorado ?

A. Raleigh

B. Harrisburg

C. Denver

D. Lincoln

–snip–

Course Outcomes: At the end of the course the student will be able to:	
22CDS46L.1	Demonstrate python proficiency in handling conditions, loops and creation of functions.
22CDS46L.2	Implement applications using lists and tuples methods.
22CDS46L.3	Implement applications using dictionary methods.

22CDS46L.4	Implement String manipulation-based applications
22CDS46L.5	Design file management system using common file operations
22CDS46L.6	Design and develop python based real world applications

Sl. No.	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Automate the Boring Stuff with Python	Al Sweigart	No Starch Press	1st Edition 2015

Web links and Video Lectures (e-Resources):

- Al Sweigart, “Automate the Boring Stuff with Python”, 2nd Edition, No Starch Press, 2020. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
- https://www.python.org
- https://developers.google.com/edu/python
- https://www.learnpython.org

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**	**PS O1**	**PS O2**
22CDS46L.1													2	
22CDS46L.2			2										2	
22CDS46L.3						2							2	
22CDS46L.4													2	
22CDS46L.5													2	
22CDS46L.6	3	3	3	3							2	2	2	

1: Low 2: Medium 3: High

Universal Human Values- II

Course Code	**22UHV47**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	**02**	Exam Hours	02

Course Learning Objectives:

This introductory course input is intended:

1. To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
2. 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 Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement toward value-based living in a natural way.
3. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature.

Module-1 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 Fulfill the Basic Human Aspirations.

Activities: Sharing about Oneself, Exploring Human Consciousness and Exploring Natural Acceptance. **5 Hours**

Module-2 – Harmony in the Human Being

Understanding Human beings 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.

Activities: Exploring Sources of Imagination in the Self, Exploring Harmony of Self with the Body and Exploring the difference of Needs of Self and Body. **5 hours**

Module 3 – Harmony in the Family and Society

Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in the Society, Vision for the Universal Human Order.

Activities: Exploring the Feeling of Trust, Exploring the Feeling of Respect and Exploring the Feeling systems to fulfil Human Goal. **5 hours**

Module-4 – 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.

Activities: Exploring the Four Orders of Nature and Co-existence in Existence. **5 hours**

Module-5 – 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

Activities: Exploring Ethical Human Conduct**,** Humanistic Models in Education and steps of Transition towards Universal Human Order. **5 hours**

Course Outcomes: At the end of the course the student will be able to:	
22UHV47.1	Practice the method of self-exploration to understand the basic human aspiration.
22UHV47.2	Distinguish between needs of self and body.
22UHV47.3	Evolve a program for self-regulation and health.
22UHV47.4	Differentiate between the characteristics and activities of different orders and study the mutual fulfilment among them.
22UHV47.5	Realize sustainable solutions to the problems in society and nature.
22UHV47.6	Develop competence in professional ethics and strategies for the transition towards a value-based life/profession.

Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
Textbooks				
1	Foundation Course in Human Values and Professional Ethics	R R Gaur, R Asthana, G P Bagaria	Excel Books, New Delhi	2nd Revised Edition, 2019
2	Teachers' Manual for A Foundation Course in Human Values and Professional Ethics	R R Gaur, R Asthana, G P Bagaria	Excel Books New Delhi	2nd Revised Edition, 2019
Reference Books				
1	Jeevan Vidya: Ek Parichaya	A Nagaraj	Jeevan Vidya Prakashan Amarkantak	1999
2	Human Values	A.N. Tripathi	New Age Intl. Publishers, New Delhi	2004

Additional Resources/Web links/Video Lectures

1. The Story of Stuff (Book).
2. The Story of My Experiments with Truth - by Mohandas Karamchand Gandhi
3. Small is Beautiful - E. F Schumacher.
4. Slow is Beautiful - Cecile Andrews
4. Economy of Permanence - J C Kumarappa
5. Bharat Mein Angreji Raj – Pandit Sunderlal
6. Rediscovering India - by Dharampal
7. Hind Swaraj or Indian Home Rule - by Mohandas K. Gandhi
8. India Wins Freedom - Maulana Abdul Kalam Azad
9. Vivekananda - Romain Rolland (English)
10. Gandhi - Romain Rolland (English)

11. UHV-I Teaching material (Presentations, Pre & Post Surveys etc.)
https://fdp-si.aicte-india.org/AicteSipUHV_download.php

12. Details of UHV-II: Universal Human Values – Understanding Harmony and Ethical Human Conduct
https://drive.google.com/file/d/1cznDaqDwKy__EKWmqJLWF94MeY4AXcsU/view?usp=sharing

13. Recorded FDP (Refresher 1 Part 1: Preparing to teach UHV-I in SIP)
https://www.youtube.com/watch?v=kejuD4faDDE&list=PLWDeKF97v9SOjS4RanhaYj4YLiImqm5pj&index=1

14. Resources, including the class notes and presentations
https://drive.google.com/drive/folders/1nh9m5ibEtvMyqekeiexAJtfbdNtmtt6-?usp=sharing

15. Hindi Recording of 5-day UHV FDP
https://www.youtube.com/playlist?list=PLWDeKF97v9SMRfe5PK1HPYnEcrrJOL6K7

16. English Recording of 5-day UHV FDP
https://www.youtube.com/playlist?list=PLWDeKF97v9SP7wSlapZcQRrT7OH0ZlGC4

Course Articulation Matrix

Course Outcomes (COs)	**Program Outcomes (POs)**													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22UHV47.1						3		2						
22UHV47.2						2			3					
22UHV47.3						2		3						
22UHV47.4							3							
22UHV47.5			3				2							
22UHV47.6								3				2		

1: Low 2: Medium 3: High

Biology for Engineers

Course Code	**22BFE47**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	**02**	Exam Hours	02

Course Learning Objectives:

1. To bring awareness of biological concepts to engineering students
2. To introduce the building blocks of life and their complexity
3. To encourage interdisciplinary studies and projects
4. To appreciate the discoveries that mimic nature and its working
5. To inculcate nature-inspired design and operational principles

Module-1 **5 Hours**

Basic Cell Biology: Introduction to Biology, The cell: the basic unit of life, Expression of genetic information-protein structure and function, Cell metabolism; Cells respond to their external environments, Cells grow and reproduce, Cellular differentiation.

Module-2 **5 Hours**

Biochemistry and Molecular Aspects of Life: Biodiversity-Chemical bonds in Biochemistry; Biochemistry and Human biology, Protein synthesis -DNA; RNA, Transcription and translation factors play key roles in protein synthesis, Differences between eukaryotic and prokaryotic protein synthesis, Stem cells and their applications.

Module-3 **5 Hours**

Bioinspired Engineering based on human physiology: Circulatory system (artificial heart, pacemaker, stents), Nervous system (Artificial neural network), Respiratory system, sensory system (electronic nose, electronic tongue), Visual and auditory prosthesis (Bionic eye and cochlear implant).

Module-4 **5 Hours**

Relevance of Biology as an interdisciplinary approach: Biological observation that led to major discoveries, Echolocation (ultrasonography, sonars), Photosynthesis (photovoltaic cells, bionic leaf), Bird flying (aircraft), Lotus leaf effect (Super hydrophobic and self-cleaning surfaces), Plant burrs (Velcro).

Module-5 **5 Hours**

Bioinspired Algorithms and Applications: Genetic algorithm, Gene expression modelling, Parallel Genetic Programming: Methodology, History, and Application to Real-Life Problems, Dynamic Updating DNA Computing Algorithms, Bee-Hive: New Ideas for Developing Routing Algorithms Inspired by Honey Bee Behaviour.

Course Outcomes: At the end of the course the student will be able to:

22BFE47.1	Discuss how the cell forms the basic building block of life
22BFE47.2	Distinguish between transcription and translation
22BFE47.3	Describe the role played by proteins within the cell
22BFE47.4	Analyze the role of bioinspired design in novel applications
22BFE47.5	Apply bioinspired design principles to other domains
22BFE47.6	Implement a simple genetic algorithm

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Text Books				
1	Biology for Engineers	Thyagarajan.S., Selvamurugan. N., Rajesh.MP, Nazeer RA, Richard W. Thilagaraj, Barathi.S., and Jaganthan.M.K	Tata McGraw Hill	2012
2	Molecular Biology	Robert Weaver	McGraw-Hill	5th Edition, 2012
Reference books				
1	Lewin's Genes XII	Jocelyn E. Krebs, Elliott S. Goldstein, Stephen T. Kilpatrick	Jones and Bartlett Learning	2017
2	Bioinspired Engineering	Jenkins, C.H.	Momentum Press	2012
3	Bio mimetics: Nature-Based Innovation	Yoseph Bar-Cohen	CRC Press	1st Edition, 2016
4	A Practical Guide to Bio-inspired Design	Hashemi Farzaneh, Helena, Lindemann, Udo	Springer	2019

Web links/Video Lectures/MOOCs

1. https://books.google.co.in/books?id=-2LNBQAAQBAJ&printsec=frontcover#v=onepage&q&f=false

2. https://www.aminotes.com/2017/02/biology-for-engineers-module-1-cocepts.html

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
22BFE47.1	2					1								
22BFE47.2		1				1								
22BFE47.3	2					2								
22BFE47.4		2										2		
22BFE47.5	2											2		
22BFE47.6		2										2		

1: Low 2: Medium 3: High

COMPUTATIONAL TOOLS FOR ENGINEERS			
Course Code:	**22CTE48**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	**01**	Exam Hours	02

Course Learning Objectives:

1. Apply modeling and simulation tools for a wide range of engineering problems.
2. Understand the analysis of data in Excel with statistics.
3. Use MATLAB and Simulink to perform engineering system analysis.

The engineering design process heavily relies on modeling and simulation. Modern simulation techniques enable the development of multi-physical, holistic system models that account for all system interactions. These digital models speed up the design and testing processes, saving time and money.

Module 1

Engineering Design Analysis

Need for engineering design analysis. Product and system design. Introduction to analysis parameters – stress, deformation, acceleration, internal force and stability. Static structural analysis of engineering design using finite element method (case studies). Heat transfer and fluid dynamics modeling and simulation using CFD software (case studies).

10 Hours

Module 2

Data Analysis with EXCEL

Calculate Mean, Median, Mode, Minimum, Maximum, Quartiles, Variance and Standard Deviation from some numbers. Analyze a population using data samples. Group data, build XY charts, apply Logarithmic Scale and Trend Line on a chart, forecast from some data, and calculate running averages. Normal Distribution, Exponential Distribution, Uniform Probabilities, Binomial Distribution, and Poisson Distribution.

4 Hours

Module 3

MATLAB and Simulink for Engineers

Applications of MATLAB and Simulink in electrical engineering, electrical machines and power system projects, simulation of rectifiers, inverters, choppers, and cycloconverters.

10 Hours

Course Project

Solve complex engineering problems via modeling and simulation. The project work is teamwork of 3-5 students. The goals should be clearly defined, use any software tool, and rigorous validation of the mathematical model should be done (experimental or theoretical).

Course Outcomes: At the end of the course, the student will be able to:	
22CTE48.1	Apply the Finite Element Method to solve engineering problems
22CTE48.2	Solve statistical problems using Excel
22CTE48.3	Perform system-level analysis using MATLAB and Simulink
22CTE48.4	Build mathematical models for any given engineering problem.
22CTE48.5	Demonstrate teamwork and communication skills

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Reference Books				
1	**MATLAB and Simulink for Engineers**	Agam Kumar Tyagi	Oxford University Press	2012
2	**Practical Finite Element Analysis**	Nitin S.Gokhale	Finite to Infinite	2020
3	**Excel Crash Course for Engineers**	Eklas Hossain	Springer	2021

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CTE48.1	1				1	1								
22CTE48.2		1			2				2					
22CTE48.3		1			2									
22CTE48.4					2	2								
22CTE48.5	1								2					

1: Low 2: Medium 3: High

Industry Oriented Training - Business Etiquettes

Course Code	**22ITB49A**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02

Course Learning Objectives:

1. Know the components of self-introduction
2. Develop a resume with the inclusion of core competencies
3. Involve and contribute to group discussions
4. Develop effective communication to succeed in the professional career
5. Know the etiquettes of digital communication

Module-1

Self-Introduction &Essentials of grooming

Self-Introduction: Learn the secret to introducing Yourself, Things to avoid when introducing yourself. Activity: Video record the self-introduction. **Essentials of grooming:** Creating the first impression, what does the well-dressed man wear? What does the well-dressed woman wear? Personal hygiene and habits. **4 Hours**

Module-2

Resume Writing

Purpose, Identifying Relevant Competencies, Understanding Applicant Tracking Systems, Lists of Competencies, Writing Accomplishment/ Objective Statements, Finding the Right Words- Action verbs, The Most Popular Resume Format, Other Popular Resume Formats, Do's and Don'ts. Activity: Students have to submit a copy of their resume. **4 Hours**

Module-3

Group Discussion

Types, process, Evaluation criteria, Do's and Don'ts Activity: Group discussions have to be held during the training sessions. **4 Hours**

Module-4

Communicate effectively

Build a Story, Just a Minute, Group Activities, Team building activities, Role Play, Presentation Skills. **4 Hours**

Module-5

Digital right and wrong

Virtual Communication: Agenda, being prepared, Dressing appropriately, background, Use Microphone and camera the right way, restraining from off tasks during virtual meetings, protecting confidential data during online presentations, time management. **4 Hours**

Course Outcomes: At the end of the course the student will be able to:	
22ITB49A.1	Articulate the essential components required for self-introduction in any business or a networking event and also recognize the need to dress appropriately for a successful career in the corporate
22ITB49A.2	Develop a resume inclusive of core competencies, and action verbs which are compatible with Applicant Tracking Systems
22ITB49A.3	Demonstrate the types, process and evaluation process of Group Discussion and carry out effective group discussions
22ITB49A.4	Develop skills required for effective communication
22ITB49A.5	Associate and be accustomed to the etiquette to be followed during online meetings

Sources

1. English for Common Interactions in the Workplace: Basic Level: Coursera: https://www.coursera.org/learn/english-common-interactions-workplace-basic-level
2. Personal Communication-Introduce Yourself With Confidence: https://www.udemy.com/course/how-to-introduce-yourself/
3. Professionalism, Grooming and Etiquette: https://www.edx.org/course/professionalism-grooming-and-etiquette
4. How to Write a Resume: https://www.coursera.org/learn/how-to-write-a-resume#syllabus
5. Group Discussion Strategies: https://www.udemy.com/course/group-discussion-strategies/
6. Communication Strategies for a Virtual Age: https://www.coursera.org/learn/communication-strategies-virtual-age#syllabus

References

1. https://simplifytraining.com/course/personal-hygiene-and-good-grooming/
2. https://www.udemy.com/course/group-discussion-strategies/
3. https://www.educba.com/course/group-discussion/
4. https://getrafiki.ai/meetings/rules-of-virtual-meeting-etiquette-every-sales-professional-should-follow/
5. https://thedigitalworkplace.com/articles/online-meeting-etiquette-for-attendees/
6. https://rigorousthemes.com/blog/virtual-meeting-etiquette-guidelines-ground-rules/

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22ITB49A.1									2	3		1		
22ITB49A.2										3		1		
22ITB49A.3									2	3	1	1		
22ITB49A.4									2	3	1	1		
22ITB49A.5									2	3	1	1		

1: Low 2: Medium 3: High

Industry Oriented Training - Computing Skills

Course Code	**22ITC49B**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	**-**	Exam Hours	02

Course Learning Objectives:

- Use logical conditions for problem-solving and also introduce the concepts of arrays
- Know functions, function calls, and parameter passing
- Introduce algorithms and appreciate their importance in problem-solving
- Introduce the core concepts of OOP’s
- Differentiate between front-end & back-end development and recognize the use of database management

Module-1

Introduction to computing constructs

Logical conditions: For Loops, Nested For Loops, While Loops, Do-While Loops, Nesting and Boxes, and combine/negate several logical conditions using logic operations AND, OR, and NOT.

Arrays & strings: Create arrays of characters (strings), use the null terminator, and manipulate strings. **4 Hours**

Module-2

Functions & Pointers

Introduction to Functions, Returning Data From a Function, Passing Data Into a Function, Getting Valid User Input, Changing Parameter Values, Pointer Basics, Changing the Pointed to Value, Walking an Array with Pointers, Dynamic Memory Allocation, Getting More Memory, Pointers to Structure. **4 Hours**

Module-3

Algorithm analysis

Introduction to Algorithm Analysis, Big-O, Big-O Examples, Dynamic Array Operations, Bubble Sort, Selection Sort, Insertion Sort, Recursion, Recursive Binary Search, Merge Sort.

4 Hours

Module-4

Object-oriented programming

Designing for Object-Oriented Programming, Core Concepts of OO Programming: Classes and objects, data abstraction, encapsulation, inheritance, benefits of inheritance, polymorphism, procedural and object-oriented programming paradigm. **4 Hours**

Module-5

Frontend and backend development

UI, Database management: DBMS overview, Relational Data Model and the CREATE TABLE Statement, Basic Query Formulation with SQL. **4 Hours**

Course Outcomes: At the end of the course the student will be able to:	
22ITC49B.1	Illustrate the use of logical conditions, declare and manipulate data into arrays
22ITC49B.2	Implement functions, function calls, and parameter passing
22ITC49B.3	Design, implement, and evaluate an algorithm to meet desired needs
22ITC49B.4	Describe the core concepts of OOP’s
22ITC49B.5	Recognize the concepts of front-end development and database management

Sources

1. Computational Thinking with Beginning C Programming Specialization: https://www.coursera.org/learn/simulation-algorithm-analysis-pointers?specialization=computational-thinking-c-programming#syllabus

2. Simulation, Algorithm Analysis, and Pointers: https://www.coursera.org/lecture/simulation-algorithm-analysis-pointers/big-o-examples-pdCan

3. Programming Fundamentals: https://www.coursera.org/learn/programming-fundamentals?specialization=c-programming#syllabus

4. Object-Oriented Programming Concepts: https://www.coursera.org/learn/concepts-of-object-oriented-programming#syllabus

5. Introduction to Back-End Development: https://www.coursera.org/learn/introduction-to-back-end-development

Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22ITC49B.1	2	1	1											
22ITC49B.2	2	1	1											
22ITC49B.3	1	1	2											
22ITC49B.4	2		1											
22ITC49B.5	2	1	1											

1: Low 2: Medium 3: High

***

Core Values of the Institution

SERVICE

A Josephite will keep service as the prime goal in everything that is undertaken. Meeting the needs of the stakeholders will be the prime focus of all our endeavors.

EXCELLENCE

A Josephite will not only endeavor to serve, but serve with excellence. Preparing rigorously to excel in whatever we do will be our hallmark.

ACCOUNTABILITY

Every member of the SJEC Family will be guided to deliver on assurances given within the constraints set. A Josephite will always keep budgets and deadlines in mind when delivering a service.

CONTINUOUS ADAPTATION

Every member of the SJEC Family will strive to provide reliable and continuous service by adapting to the changing environment.

COLLABORATION

A Josephite will always seek to collaborate with others and be a team-player in the service of the stakeholders.

Objectives

- Provide Quality Technical Education facilities to every student admitted to the College and facilitate the development of all round personality of the students.
- Provide most competent staff and excellent support facilities like laboratory, library and internet required for good education on a continuous basis.
- Encourage organizing and participation of staff and students in in-house and outside Training programmes, seminars, conferences and workshops on continuous basis.
- Provide incentives and encouragement to motivate staff and students to actively involve in research-innovative projects in collaboration with industry and R &D centres on continuous basis
- Invite more and more number of persons from industry from India and abroad for collaboration and promote Industry-Institute Partnership.
- Encourage consultancy and testing and respond to the needs of the immediate neighbourhood.

St Joseph Engineering College

AN AUTONOMOUS INSTITUTION
Affiliated to VTU, Belagavi | Recognised by AICTE, New Delhi
Accredited by NAAC with A+ Grade
B.E. (CSE, ECE, EEE, ME, CIV), MBA & MCA Accredited by NBA, New Delhi

Vamanjoor, Mangaluru - 575 028, Karnataka, India
Ph: 91-824-2868100 / 2263753 / 54 / 55
E-mail: sjec@sjec.ac.in| Website: www.sjec.ac.in