

# BE SCHEME & SYLLABUS

## Second Year (III and IV Semester)

With effect from 2022-23

Computer Science  
and Engineering



**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 and Engineering**

#### **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

SI No	SUBJECTS	Page No
1	Department Vision, Mission, Program Educational Objectives (PEOs)	04
2	Program Outcomes POs and Program Specific Outcomes PSOs	05
3	Scheme – III Semester Computer Science and Engineering	06
4	Scheme – IV Semester Computer Science and Engineering	07
<b>III Semester</b>		
5	22CSE31 - Mathematical Foundations for Computer Science	09
6	22CSE32 – Data Structures and Applications (Integrated Course)	11
7	22CSE33 - Digital Principles and Design (Integrated Course)	15
8	22CSE34 - Computer Organization and Architecture	18
9	22CSE351 - UNIX Shell Programming	20
10	22CSE352 - Dot Net Programming	22
11	22CSE36L - Object Oriented Programming with Java Laboratory	24
12	22UHV37 - Universal Human Values- II	27
13	22BFE37 - Biology for Engineers	30
14	22IEP38 - IoT Enabled Prototyping	32
15	22ITB39A - Industry Oriented Training - Business Etiquettes	34
16	22ITC39B - Industry Oriented Training - Computing Skills	36
<b>IV Semester</b>		
17	22CSE41 - Discrete Mathematical Structures	39
18	22CSE42 - Design and Analysis of Algorithms (Integrated Course)	41
19	22CSE43 - Database Management System (Integrated Course)	44
20	22CSE44 - Operating System	48
21	22CSE451 - Data Analysis using R Programming	50
22	22CSE452 – Software Testing	52
23	22CSE46L - Python Programming Laboratory	54
24	22UHV47 - Universal Human Values- II	57
25	22BFE47 - Biology for Engineers	60
26	22CTE48 - Computational Tools for Engineers	62
27	22ITB49A - Industry Oriented Training - Business Etiquettes	64
28	22ITC49B - Industry Oriented Training - Computing Skills	66

## **ABOUT THE DEPARTMENT**

The Department of Computer Science and Engineering was setup during the inception of the college in 2002. The primary objective of this program is to prepare students for successful careers in Computer and Information technology industry that meet the needs of Indian and multinational organizations. The Department started Research program in the year 2011. Currently, the student intake is 180 at the UG level. The program involves wide variety of courses which enable the students to formulate, solve and analyze computer engineering problems, prepare them for graduate studies and develop the ability to synthesize data and technical concepts for application design & implementation of real time software products. The faculty of the Department are actively involved in teaching and research with specializations in Cloud Computing, Image Processing, Process Mining, Natural Language Processing and Soft & Evolutionary Computing. The Department was accredited by NBA for 2 years from June 2013 to May 2015, for six years from July 2016 to June 2022 and is reaccredited by NBA for three years from July 2022 to June 2025. The Department has received a grant of Rs. 19 lakhs from AICTE for setting up Center of Excellence in Augmented Reality and Virtual Reality (AR/VR) under MODROBS 2020-21 scheme.

### **DEPARTMENT VISION**

- To be recognized as a centre of excellence in computer and allied areas with quality learning and research environment.

### **DEPARTMENT MISSION**

- Prepare competent professionals in the field of computer and allied fields enriched with ethical values.
- Contribute to the socio-economic development of the country by imparting quality education in Computer and Information Technology.
- Enhance employability through skill development.

### **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

PEO 1: Apply Engineering Fundamentals: To impart to students a sound foundation and ability to Apply engineering fundamentals, mathematics, science and humanities necessary to formulate, analyze, design and implement engineering problems in the field of computer science.

PEO 2: Work in CS and allied fields: To develop in students the knowledge of fundamentals of computer science and engineering to work in various related fields such as network, data, web and system engineering.

PEO 3: Teamwork: To develop in students the ability to work as a part of team through effective communication on multidisciplinary projects.

PEO 4: Successful Career: To train students to have successful careers in computer and information technology industry that meets the needs of society enriched with professional ethics.

PEO 5: Higher Education: To develop in students the ability to pursue higher education and engage in research through continuous learning.

## 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)

Graduates of the Computer Science and Engineering program are able to

- PSO 1.** Understand the principles underlying entrepreneurship, freelancing and the requirements to initiate a start-up in the IT or related domains.
- PSO 2.** Participate effectively in competitive examinations for career growth, higher studies and to pursue research.

### III Semester (B.E. - CSE Engineering)

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	BSC	22CSE31	Mathematical Foundations for Computer Science	MAT	MAT	2	2	-	03	50	50	100	3
2	IPCC	22CSE32	Data Structures and Applications (Integrated)	CSE	CSE	2	2	2	03	50	50	100	4
3	IPCC	22CSE33	Digital Principles and Design (Integrated)	CSE	CSE	3	-	2	03	50	50	100	4
4	PCC	22CSE34	Computer Organization and Architecture	CSE	CSE	3	-	-	03	50	50	100	3
5	ESC	22CSE35X	ESC/ETC/PLC	CSE	CSE	3	-	-	03	50	50	100	3
6	PCCL	22CSE36L	Object Oriented Programming with Java Laboratory	CSE	CSE	-	-	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	-
<b>Total</b>						15	4	10	24	<b>450</b>	<b>400</b>	<b>850</b>	<b>21</b>

22CSE35X : Engineering Science Course/Emerging Technology Course/Programming Language Course

22CSE351      UNIX Shell Programming

22CSE352

Dot Net Programming



**IV Semester (B.E. - CSE Engineering)**

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	BSC	22CSE41	Discrete Mathematical Structures	MAT	MAT	2	2	-	03	50	50	100	3
2	IPCC	22CSE42	Design and Analysis of Algorithms (Integrated)	CSE	CSE	2	2	2	03	50	50	100	4
3	IPCC	22CSE43	Database Management System (Integrated)	CSE	CSE	3	-	2	03	50	50	100	4
4	PCC	22CSE44	Operating System	CSE	CSE	3	-	-	03	50	50	100	3
5	ESC	22CSE45X	ESC/ETC/PLC	CSE	CSE	3	-	-	03	50	50	100	3
6	PCCL	22CSE46L	Python Programming Laboratory	CSE	CSE	-	-	2	03	50	50	100	1
7	HSMC	22UHV47	Universal Human Values – II	COM	COM	2	-	-	02	50	50	100	2
	HSMC	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	-
<b>Total</b>						15	4	10	24	<b>450</b>	<b>400</b>	<b>850</b>	<b>21</b>

22CSE45X : Engineering Science Course/Emerging Technology Course/Programming Language Course			
22CSE451	Data Analysis using R Programming	22CSE452	Software Testing

# **III Semester**

<b>Mathematical Foundations for Computer Science</b>			
Course Code	<b>22CSE31</b>	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
<b>Course Learning Objectives:</b>			
<ul style="list-style-type: none"> <li>To apply the least square method numerically to find the curve of best fit and to analyze the data using Correlation and Regression.</li> <li>To apply probability theory and random processes that serve as an essential tool for applications of Computer Science Engineering.</li> <li>To implement principles of advanced engineering mathematics through linear algebra.</li> </ul>			
<b>Module-1 (8 hours)</b>			
<b>Statistical Methods and Curve Fitting:</b> Correlation and Regression-Karl Pearson's coefficient of correlation-problems. Regression analysis- lines of regression -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 $y = ax^b$			
<b>Module-2 (8 hours)</b>			
<b>Probability and Discrete Probability Distributions: Probability</b> – Introduction, Conditional probability and Baye’s Theorem, Discrete Random variables, probability mass/density functions, cumulative density function. Binomial, Poisson, distributions (No derivation for mean and standard deviation).			
<b>Module-3 (8 hours)</b>			
<b>Continuous Probability Distributions and Joint Probability Distribution:</b> Continuous Random variables, probability mass/density functions, cumulative density function. exponential and normal distributions- problems (No derivation for mean and standard deviation) Joint distribution of random variables – Expectation, Covariance and Correlation			
<b>Module-4 (8 hours)</b>			
<b>Sampling theory:</b> 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.			
<b>Module-5 (8 hours)</b>			
<b>Linear Algebra:</b> Inner product, length & orthogonality, orthogonal set, orthogonal projection Gram-Schmidt process, QR factorization of matrices, Eigenvalues and Eigenvectors (Recapitulation). The singular value decomposition.			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE31.1</b>	Apply the Correlation and regression analysis to fit a suitable mathematical model for the data
<b>22CSE31.2</b>	Apply the least square methods to fit a suitable curve for given data
<b>22CSE31.3</b>	Analyze the probability models arising in Engineering field using discrete and continuous probability distributions.
<b>22CSE31.4</b>	Construct Joint Probability Distributions and apply the knowledge in engineering problems for feasible random events
<b>22CSE31.5</b>	Use sampling theory in the study of samples
<b>22CSE31.6</b>	Apply the technique linear algebra in computer applications.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Higher Engineering Mathematics	B.S. Grewal	Khanna Publishers	44 <sup>th</sup> Edition, 2017
2	Linear Algebra & its applications	David C. Lay	Pearson Publication	3 <sup>rd</sup> Edition, 2014
3	Probability & Statistics for Engineers & Scientists	Ronald Walpole, R H Myers, S L Myers and K. Ye	Pearson Publication	9 <sup>th</sup> Edition, 2016
<b>Reference Books</b>				
1	Advanced Engineering Mathematics	C.Ray Wylie, Louis C.Barrett	McGraw- Hill Book Co., New York,	6 <sup>th</sup> Edition, 2003
2	Linear Algebra & its applications	Gilbert Strang	Cengage Learning India Edition,	4 <sup>th</sup> Edition 2006
3	Digital signal processing – Principles Algorithms, and Applications	Proakis & Manolakis	Pearson Education	4th Edition,2007

**Web links and Video Lectures (e-Resources):**

- <https://youtu.be/0VTapKh3qBw>
- <https://youtu.be/-UJr1XjyfME>
- <https://youtu.be/VWlqpstJ6Mc>
- <https://youtu.be/riXKFISI-Kk>
- <https://youtu.be/6XlrvowuXdA>
- <https://youtu.be/WkDxhfxLf-M>
- <https://youtu.be/qAHEuO3u4Cg>
- <https://youtu.be/V7BtsK6WKE4>

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE31.1	2										1			
22CSE31.2		3												
22CSE31.3		3												
22CSE31.4	2										1			
22CSE31.5	2										1			
22CSE31.6		2												

1: Low 2: Medium 3: High

<b>Data Structures and Applications</b>			
Course Code	<b>22CSE32</b>	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
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• Explain fundamentals of data structures and applications that are essential for</li> <li>• Programming and problem solving.</li> <li>• Analyze linear and non-linear data structures.</li> <li>• Design and develop various basic and advanced data structures.</li> <li>• Demonstrate sorting and searching algorithms.</li> <li>• To understand the basic concepts of hashing.</li> </ul>			
<b>Module-1 Basics of DS and Strings (8 hours)</b>			
<p><b>Basic Concepts:</b> Basic terminology, Classifications (Primitive &amp; Non-Primitive), Data Structure Operations, Structures, Nested Structures, Array of Structures, Self-Referential Structures, Unions, Dynamic Memory Allocation Functions</p> <p><b>Strings:</b> Reading Strings, Writing Strings, Operations on Strings, Array of Strings, Pointers and Strings</p> <p><b>Text Book 1: Chapter 2.1, 2.2, 2.3, 5.1, 5.2, 5.3, 5.5, 5.6, 4.1, 4.2, 4.3, 4.4</b></p>			
<b>Module-2 Stacks and Queues (8 hours)</b>			
<p><b>Stacks:</b> Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. Recursion.</p> <p><b>Queues:</b> Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeue, Priority Queues.</p> <p><b>Text Book 1: Chapter 7.7, 8.4</b></p> <p><b>Text Book 2: Chapter 3.1, 3.2, 3.3, 3.4, 3.6</b></p>			
<b>Module-3 Linked List and Graphs (8 hours)</b>			
<p><b>Linked Lists:</b> Definition, Representation of linked lists in Memory,</p> <p><b>Single Linked list operations:</b> Traversing, Searching, Insertion, Updation and Deletion,.</p> <p><b>Doubly Linked list operations:</b> Traversing, Searching, Insertion, Updation and Deletion.</p> <p><b>Applications of Linked lists:</b> Polynomials, Sparse matrix representation. Programming Examples</p> <p><b>Text Book 1: Chapter 6.1, 6.2, 6.3, 6.4, 6.5.</b></p> <p><b>Text Book 2: Chapter 4.4, 4.7</b></p>			
<b>Module-4 Trees (8 hours)</b>			
<p><b>Trees:</b> Basic terminology, Types of trees, Traversing the binary tree, Binary Search Trees, Operations on Binary Search Trees, Threaded Binary Trees.</p> <p><b>Graphs:</b> Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search.</p> <p><b>Text Book 1: Chapter 9.1, 9.2, 9.4, 10.1 - 10.3, 13.1, 13.2, 13.3, 13.5, 13.6</b></p>			
<b>Module-5 Searching and Hashing (8 hours)</b>			
<p><b>Searching and Sorting:</b> Jump Search, Insertion sort, Radix Sort, Shell Sort.</p> <p><b>Hashing and Collision:</b> Introduction, Hash Tables, Hash Functions, Different Hash Functions, Collisions, Pros and Cons of Hashing, Applications of Hashing.</p> <p><b>Text Book 1: Chapter 14.1, 14.5, 14.8, 14.12, 14.14, 15.1-15.7</b></p>			

## **PRACTICAL MODULE**

### **A–Demonstration (offline/virtual):**

- A1. Create a menu driven program for displaying employee details using structure and Union
- A2. Write a C program to demonstrate the operations of String
- A3. Write a C program to demonstrate the operations of Stack
- A4. Write a C program to demonstrate addition and subtraction of Polynomials

### **B–Exercise (compulsorily to be conducted):**

- B1. 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.
- B2. 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 onto 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.

- B3. Design, Develop and Implement a menu driven Program in C for the following operations on the Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, Ph No.
  - 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.
  - e. Exit.
- B4. 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.
  - a. Use Hash function  $H(K)=K \text{ mod } m$  (remainder method), and implement hashing technique to map a given key K to the address space L.
  - b. Resolve the collision (if any) using linear probing.
- B5. 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.
  - e. Exit.

### **C–Structured Enquiry (compulsorily any 4 to be conducted):**

- C1. Write a C program to implement Radix sort algorithm for sorting a given list of integers in ascending order.
- C2. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Student Data with the fields: USN, Name, Branch, Sem, Ph No.
  - a. Create a DLL of N Students Data by using front insertion.
  - b. Display the status of DLL and count the number of nodes in it.
  - c. Perform Insertion of Double Linked List.
  - d. Perform Deletion of Doubly Linked List.
  - e. Exit.
- C3. Design, Develop and Implement a menu driven Program for the following.
  - a. Demonstrate dynamic allocation of 2D array of integers

<p>b. Read <math>m \times n</math> sparse matrix into an array.</p> <p>c. Compute transpose of <math>m \times n</math> sparse matrix</p> <p>d. Display sparse matrix.</p> <p>Support the program with functions for each of the above operations.</p> <p>C4. Design, Develop and Implement a Program for the following operations on Strings.</p> <p>a. Read a main String (STR) and a Pattern String (PAT).</p> <p>b. Implement any algorithm to Perform Pattern Matching Operation.</p> <p>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.</p> <p>C5. Design, Develop and Implement a program for the following operations on Graph:</p> <p>a. Create any Graph of N cities using adjacency matrix</p> <p>b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method</p> <p style="text-align: center;"><b>D–Open Ended Experiments (any two):</b></p> <p>D1. Implement various Sorting techniques</p> <p>D2. Implement various Searching techniques</p>
---

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE32.1</b>	Apply data structures (pointers, arrays, structures and strings) for data organization and traversal.
<b>22CSE32.2</b>	Analyze and implement sorting, searching and data organization using the data structures Stacks, Queues and Linked Lists.
<b>22CSE32.3</b>	Apply trees and graphs for data ordering, data searching and evaluating expressions.
<b>22CSE32.4</b>	Demonstrate the operations on Hash tables and explain various collision techniques
<b>22CSE32.5</b>	Illustrate various searching and sorting algorithms.
<b>22CSE32.6</b>	Apply data structures to implement 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
<b>Textbooks</b>				
1	Data Structures using C	Reema Thareja	Oxford press	2 <sup>nd</sup> Edition, 2014
2	Fundamentals of Data Structures in C	Ellis Horowitz and Sartaj Sahni,	Universities Press	2 <sup>nd</sup> Edition, 2008
<b>Reference Books</b>				
1	Data Structures using C	Aaron M.Tenenbaum, Yedidyah Langsam, Moshe J.Augenstein	Pearson Education	1 <sup>st</sup> Edition, 2009
2	Data Structures: A Pseudocode Approach with C	Richard F. Gilberg and Behrouz A. Forouzan	Cengage Learning	2 <sup>nd</sup> Edition, 2005

**Web links and Video Lectures (e-Resources):**

- **Linked List:**  
[https://www.academia.edu/42067890/Data\\_Structures\\_Using\\_C\\_2e\\_Reema\\_Thareja](https://www.academia.edu/42067890/Data_Structures_Using_C_2e_Reema_Thareja) Overall Concepts
- **NPTEL:** Computer Science and Engineering – Data Structures And Algorithms
- **Stacks and Queues:** <https://www.simplilearn.com/tutorials/data-structure-tutorial/stacks-and-queues>

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE32.1						2					2			
22CSE32.2									2					2
22CSE32.3											2			2
22CSE32.4						1					1			
22CSE32.5						2			3		2		1	3
22CSE32.6						2								3

1: Low 2: Medium 3: High



<b>Digital Principles and Design</b>			
Course Code	<b>22CSE33</b>	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
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• Make use of basic gates and design the logic circuits.</li> <li>• Apply the simplifying techniques in the design of combinational circuits</li> <li>• Differentiate the combinational and sequential circuits</li> <li>• Demonstrate the use of flip-flops in the construction of registers and counters</li> <li>• Illustrate how to write simple HDL programs which describe the digital circuits</li> </ul>			
<b>Module-1: Digital Logic and Principles of combination logic (8 Hours)</b>			
<p><b>Digital Logic and Principles of combination logic:</b> 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.</p> <p><b>Text Book 1: Chapter 2.1, 2.2, 2.4, 3.1, 3.2, 3.7, 3.8</b></p>			
<b>Module-2: Combinational logic circuit design (8 Hours)</b>			
<p><b>Combinational logic circuit design:</b> Karnaugh map simplification, Don't care conditions, Simplification by Quine Mc-Cluskey 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.</p> <p><b>Text Book 1: Chapter 3.3-3.6, 3.9, 3.11, 3.12, 3.13</b></p>			
<b>Module-3: Data Processing circuits (8 Hours)</b>			
<p><b>Data Processing circuits :</b> 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.</p> <p><b>Text Book 1: Chapter 4.1-4.2, 4.4-4.12, 4.14</b></p>			
<b>Module-4: Latches and Flip-Flops (8 Hours)</b>			
<p><b>Latches and Flip-Flops:</b> 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.</p> <p><b>Text Book 1: Chapter 8.1-8.5, 8.8-8.11</b></p>			
<b>Module-5: Registers and Counters (8 Hours)</b>			
<p><b>Registers and Counters:</b> 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.</p> <p><b>Text Book 1: Chapter 9.1, 9.7, 9.8, 10.1-10.7</b></p> <p><b>Text Book 2: Chapter 13.1</b></p>			

<p><b>PRACTICALMODULE</b></p> <p><b>A–Demonstration (offline/virtual):</b></p> <p>1. Design and implementation of a Half adder, Half Subtractor and a Full Adder using basic gates. Implement Full Subtractor in Verilog HDL.</p>
---

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 the 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 ( $n \leq 9$ ) and demonstrate on a 7-segment display (using IC-7447).
  6. Design and Testing Shift Register/Ring Counter/Johnson Counter
- 7. Open ended experiment covering the concept of entire syllabus**
- Design and Testing Sequence Generator
  - Use Universal gates and IC's for code conversion and arithmetic Operations
  - Design and Verify on Different Counters.

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

<b>22CSE33.1</b>	Explain the usage of basic gates, universal gates, and Boolean laws in designing various digital circuits
<b>22CSE33.2</b>	Apply the simplification techniques like Karnaugh map and Quine Mc-Clusky to design various combinational circuits.
<b>22CSE33.3</b>	Describe the operation and design of various data processing circuits and implement multiplexer's circuit.
<b>22CSE33.4</b>	Identify the various types of flip-flops and use them in the design of Registers and Counters
<b>22CSE33.5</b>	Differentiate between Moore and Mealy model and construct different types of counters using these models.
<b>22CSE33.6</b>	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
<b>Textbooks</b>				
1	Digital Principles and Application	Donald P Leach, Albert Paul Malvino & Goutam Saha	Tata McGrawHill	8 <sup>th</sup> Edition, 2017
2	Fundamentals of Logic Design	Charles H Roth and Larry L Kinney	Cengage Learning	7 <sup>th</sup> Edition, 2013
<b>Reference Books</b>				
1	Digital Design Principles and Practices	John F. Wakerly	Pearson Education	4 <sup>th</sup> Edition, 2008
2	Digital Design with an Introduction to the Verilog HDL	M. Morris Mano, Michael D. Ciletti	Pearson Education	5 <sup>th</sup> Edition, 2013

**Web links and Video Lectures (e-Resources):**

- [https://onlinecourses.nptel.ac.in/noc22\\_ee110/preview](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
22CSE33.1	3	1												1
22CSE33.2	1	2	2											1
22CSE33.3	3		1											1
22CSE33.4		1		2										1
22CSE33.5		1		2										1
22CSE33.6		1			2									1

1: Low 2: Medium 3: High

<b>Computer Organization and Architecture</b>			
Course Code	<b>22CSE34</b>	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	Credits	03
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• To explain the basic subsystems of a computer, their organization, structure and operations</li> <li>• Illustrate the concept of programs as sequences of machine instructions</li> <li>• Demonstrate different ways of communicating with I/O devices and standard I/O interfaces</li> <li>• Describe arithmetic operations with integer operands.</li> <li>• Exposure to Computer Architecture and instruction level parallelism</li> </ul>			
<b>Module-1 Basic Structure of Computers and Machine Instructions and Programs (8 hours)</b>			
<p><b>Basic Structure of Computers:</b> Basic Operational Concepts, Bus Structures, Performance-Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement.</p> <p><b>Machine Instructions and Programs:</b> Memory Location and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Subroutines and use of stack frames.</p> <p><b>Text book 1: Chapter 1 – 1.3, 1.4, 1.6 (1.6.1 to 1.6.4, 1.6.7), Chapter 2 – 2.2 to 2.6, 2.8 to 2.9</b></p>			
<b>Module-2 Input/Output Organization (8 hours)</b>			
<p><b>Input/Output Organization:</b> 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.</p> <p><b>Text book 1: Chapter 2 – 2.7, Chapter 4 – 4.1, 4.2 (4.2.1 to 4.2.5), 4.4, 4.5</b></p>			
<b>Module-3 Memory unit and Multiprocessor Architecture (8 hours)</b>			
<p><b>Memory unit:</b> Memory Hierarchy, Basics of Cache memory, Cache mapping techniques, Hit Rate and Miss Penalty.</p> <p><b>Multiprocessor Architecture:</b> Centralized shared-memory architecture, Distributed memory architecture, <b>Basic scheme for enforcing coherence:</b> Snooping coherence protocols, Basic implementation techniques.</p> <p><b>Text book 1: Chapter 5 – 5.4 to 5.5 (5.5.1 to 5.5.2), 5.6 (5.6.2)</b>  <b>Text book 2: Chapter 5 – 5.1, 5.2, 5.4</b></p>			
<b>Module-4 Integer arithmetic and The Processor (8 hours)</b>			
<p><b>Overflow in integer arithmetic:</b> Numbers, Arithmetic operations and characters</p> <p>Arithmetic unit: Multiplication of two numbers, A signed operand multiplication, Booth algorithm, Bit pair recoding and CSA – integer division.</p> <p><b>The Processor:</b> Processing unit: Fundamental concepts, Execution of complete instruction, Multiple bus organization.</p> <p><b>Text book 1: Chapter 2- 2.1, Chapter 6 – 6.3 to 6.6, Chapter 7- 7.1 to 7.3</b></p>			
<b>Module-5 Instruction level parallelism and Pipelining (8 hours)</b>			
<p><b>Instruction level parallelism:</b> Introduction and challenges, Data dependencies and Hazards: Data dependencies, Name dependencies, Control Dependences</p> <p><b>Pipelining:</b> Introduction, A simple implementation of a RISC instruction set, The classic five-stage pipeline for a RISC processor, Basic performance issues in pipelining.</p> <p><b>Text book 2: Chapter 3 – 3.1, Appendix C</b></p>			
<b>Course Outcomes:</b> At the end of the course the student will be able to:			
<b>22CSE34.1</b>	Describe computer hardware and the basic functionality, interconnection, addressing techniques and instruction sequencing.		

<b>22CSE34.2</b>	Explain the basics concepts of I/O, interrupts, direct memory access technique and types of buses.
<b>22CSE34.3</b>	Illustrate cache memory mapping techniques, various memory architectures and protocols for cache coherence.
<b>22CSE34.4</b>	Describe different algorithms used to perform arithmetic operations.
<b>22CSE34.5</b>	Illustrate organization of a processor with single and multiple bus for instruction execution.
<b>22CSE34.6</b>	Examine 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
<b>Textbooks</b>				
1	Computer Organization	C Hamacher, Z Vranesic	Tata McGraw Hill	5 <sup>th</sup> Edition, 2011
2	Computer Architecture A Quantitative Approach	John L Hennessy, David A Patterson	Elsevier	5 <sup>th</sup> Edition 2012
<b>Reference Books</b>				
1	Computer Organization and Design	David A. Patterson, John L. Hennessy	M.K Publishers	4 <sup>th</sup> Edition, 2010
2	Computer Organization and Architecture	William Stallings	Pearson	9 <sup>th</sup> Edition, 2014
3	Computer Organization and Design MIPS Edition	Patterson	Elsevier;	6 <sup>th</sup> Edition, 2021
4	Computer Organization and Architecture	J. S. Katre , Harish G, Narula, Khushboo Shah	TechKnowledge Publications	2020

**Web links and Video Lectures (e-Resources):**

- <https://www.coursera.org/learn/comparch>
- <https://nptel.ac.in/courses/106103068>
- <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
<b>22CSE34.1</b>	2	1	1	1	1							1		2
<b>22CSE34.2</b>	1		1							1		1		1
<b>22CSE34.3</b>	1	1	1			2						1		1
<b>22CSE34.4</b>	1	1	1	1								1		2
<b>22CSE34.5</b>		1	1	1			1				1	1		
<b>22CSE34.6</b>	1	1		1	1								1	

1: Low 2: Medium 3: High

<b>UNIX Shell Programming</b>			
Course Code	<b>22CSE351</b>	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	Credits	03
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• To Familiarize student's effective use of Unix concepts, commands, and terminology</li> <li>• Identify, access, and evaluate UNIX file system</li> <li>• Understand UNIX command syntax and semantics.</li> <li>• Ability to read and understand specifications, scripts, and programs.</li> <li>• Analyze Facility with UNIX Process.</li> </ul>			
<b>Module-1 Introduction (8 hours)</b>			
<p><b>Introduction of UNIX</b> - Introduction, History, Architecture, Experience the Unix environment, Basic commands ls, cat, cal, date, calendar, who, printf, tty, sty, uname, passwd, echo, tput, and bc.  <b>Textbook 1: Chapter 1,2,3.</b></p>			
<b>Module-2 UNIX File System (8 hours)</b>			
<p><b>UNIX File System-</b> The file, what's in a filename? The parent-child relationship, pwd, the Home directory, absolute pathnames, using absolute pathnames for a command, cd, mkdir, rmdir, Relative pathnames, The UNIX file system, Commands to handle ordinary files.  <b>Textbook 1: Chapter 4, Chapter 5.1-5.4, 5.8-5.12</b></p>			
<b>Module-3 File Attributes, File Permissions (8 hours)</b>			
<p><b>Basic File Attributes</b> - ls -l, the -d option, File Permissions, chmod, Security and File Permission, users and groups, security level, changing permission, user masks, changing ownership and group, File Attributes, More file attributes: hard link, symbolic link, umask, find. Simple filters.  <b>Textbook 1: Chapter 6,11,12</b></p>			
<b>Module-4 Shell Scripts (8 hours)</b>			
<p><b>Introduction to the Shell Scripting</b> - Introduction to Shell Scripting, Shell Scripts, read, Command Line Arguments, Exit Status of a Command, The Logical Operators &amp;&amp; and   , exit, if, and case conditions, expr, sleep and wait, while, until, for, \$, @, redirection. The here document, set, trap Sample Validation and Data Entry Scripts, The Shell Interpretive cycle, Pattern matching, Escaping and quoting, Three standard files, Special files, pipes, command substitution, shell variables.  <b>Textbook 1: Chapter 8, 14</b></p>			
<b>Module-5 UNIX Process(8 hours)</b>			
<p>Introduction to UNIX System process: Mechanism of process creation. Parent and child process. The ps command with its options. Executing a command at a specified point of time: at command. Executing a command periodically: cron command and the crontab file, job control.  <b>Textbook 1: Chapter 9</b></p>			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE351.1</b>	Understand the basics of Unix concepts and commands
<b>22CSE351.2</b>	Evaluate the UNIX file system
<b>22CSE351.3</b>	Apply Changes in file system
<b>22CSE351.4</b>	Understand scripts and programs.
<b>22CSE351.5</b>	Apply scripting knowledge to write programs
<b>22CSE351.6</b>	Analyze Facility with UNIX system process

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	UNIX – Concepts and Applications	Sumitabha Das	Tata McGraw Hill	4 <sup>th</sup> Edition, 2006
<b>Reference Books</b>				
1	Unix Shell Programming	Yashwant Kanetkar	BPB Publications	1 <sup>st</sup> Edition, 2003
2	Introduction to UNIX	M G Venkatesh Murthy	Pearson	1 <sup>st</sup> Edition, 2005

**Web links and Video Lectures (e-Resources):**

- <https://www.youtube.com/watch?v=ffYUfAqEamY> 2.
- <https://www.youtube.com/watch?v=Q05NZiYFcD0> 3.
- <https://www.youtube.com/watch?v=8GdT53KDIyY> 4.
- <https://www.youtube.com/watch?app=desktop&v=3Pga3y7rCgo>

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE351.1		2	2									2		
22CSE351.2		2	2									2		
22CSE351.3		2	2									2		
22CSE351.4		2	2									2		
22CSE351.5		2	2									2		
22CSE351.6		2	2									2		

1: Low 2: Medium 3: High

<b>Dot Net Programming</b>			
Course Code	<b>22CSE352</b>	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	Credits	03
<b>Course Learning Objectives:</b> The objective of the course is to <ul style="list-style-type: none"> <li>• Learn the syntax and semantics of C#</li> <li>• Understand Object Oriented Programming concepts in C#.</li> <li>• Interpret Interfaces and define custom interfaces for applications.</li> <li>• Implement properties, indexers, generics and collections in C#</li> </ul>			
<b>Module-1 Introducing Microsoft Visual C# (8 hours)</b>			
<b>Introducing Microsoft Visual C#:</b> Working with variables, operators and expressions, Writing methods and applying scope, Using decision statements <b>Text Book 1: Chapter 2 - 4</b>			
<b>Module-2 Understanding loops, exceptions and the C# object model (8 hours)</b>			
Using compound assignment and iteration statements, Managing errors and exceptions, Creating and Managing classes and objects <b>Text Book 1: Chapter 5 - 7</b>			
<b>Module-3 Arrays and Parameters (8 hours)</b>			
Understanding values and references, Creating value types with enumerations and structures, Using arrays, Understanding parameter arrays <b>Text Book 1: Chapter 8 - 11</b>			
<b>Module-4 Inheritance and Resource Management (8 hours)</b>			
Working with inheritance, Creating interfaces and defining abstract classes, Using garbage collection and resource management <b>Text Book 2: Chapter 12 - 14</b>			
<b>Module-5 Defining Extensible Types with C# (8 hours)</b>			
Implementing properties to access fields, Using indexers, Introducing generics (up to Generics and constraints), Using collections (up to The Stack <T> collection class) <b>Text Book 3: Chapter 15 - 18</b>			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE352.1</b>	Build applications on Visual Studio .NET platform by understanding the syntax and semantics of C#
<b>22CSE352.2</b>	Manage errors and handle exceptions in C#
<b>22CSE352.3</b>	Demonstrate Object Oriented Programming concepts in C# programming language
<b>22CSE352.4</b>	Design custom interfaces for applications and leverage the available built-in interfaces in building complex applications.
<b>22CSE352.5</b>	Demonstrate garbage collection and resource management in C#
<b>22CSE352.6</b>	Illustrate the use of properties, indexers, generics and collections in C#



Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Microsoft Visual C# Step by Step	John Sharp,	PHI Learning Pvt. Ltd.	8 <sup>th</sup> Edition, 2016
<b>Reference Books</b>				
1	C# 6 and .NET Core 1.0	Christian Nagel	Wiley India Pvt Ltd	1 <sup>st</sup> Edition, 2016
2	Essential C# 6.0	Mark Michaelis	Pearson Education India	5 <sup>th</sup> Edition, 2016

- Web links and Video Lectures (e-Resources):**
- <https://dotnet.microsoft.com/en-us/>
  - <https://www.javatpoint.com/net-framework>
  - <https://dotnettutorials.net/>
  - <https://github.com/topics/dotnet>

### Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE352.1														2
22CSE352.2														2
22CSE352.3														2
22CSE352.4														2
22CSE352.5					2								2	
22CSE352.6					2						2		2	

1: Low 2: Medium 3: High

### Object Oriented Programming with Java Laboratory

Course Code	<b>22CSE36L</b>	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	20	Credits	01

**Course Learning Objectives:** The objective of the course is to

- To build software development skills using java programming for real-world applications.
- To understand and apply the concepts of classes, Inheritance, Overloading, Overriding, packages, Interfaces, Exception handling.
- To develop applications using generic programming and event handling

#### Experiments (8 hours)

##### A–Demonstration (offline/virtual):

A1. 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, and 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

A2. Develop a java application with Employee class which includes Emp\_name, Emp\_id, Address, Mail\_id, Mobile\_no as members. Inherit the classes, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of the base class. Based on the criteria given bellow generate pay slips for the employees with their gross and net salary in the respective sub classes.

Professor: 97% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund.

Associate Professor: 70% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund

Assistant Professor: 60% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund

A3. Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes Contain only the method printArea( ) that prints the area of the given shape.

A4. Write a program to demonstrate finally block in case of :

- i) No exception
- ii) Exception
- iii) Return statement

- A5. Write a Java program to input Job applicant name, Roll looking for and the Age from the Job seeker. Throw an user-defined exception, TooEarlyToApply if the entered age < 18 or TooLateToApply if the entered age > 40.
- A6. Implement a Java program to handle the Mouse based events ( mouseClicked( ), mouseEntered( ), mouseExited( ), mousePressed( ) and mouseReleased( ) ) with appropriate display to print the entry point, exit point and click point of the Mouse.
- A7. 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.
- A8. Write a Java Program to implement producer consumer problem using inter thread communication.
- A9. a). Write a java program to Demonstrate the String Compare Methods and
1. Equals()
  2. equalIgnoreCase()
  3. startsWith()
  4. endsWith()
  5. compareTo()
- b). Using StringBuffer Class Demonstrate insert(), delete(), append(), reverse() and capacity () methods

WAPP that reads on file name from the user, then displays information about whether.

1. The file exists
2. whether the file is readable
3. whether the file is writable
4. Type of file
5. Length of the file in bytes.

### **B– Problem Based Learning:**

**Case Study:** Develop a Real Time Application using the concepts covered in the course, with suitable GUI .

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE36L.1</b>	Understand and apply the concepts of classes, packages in object-oriented programming.
<b>22CSE36L.2</b>	Understand and apply the concepts of polymorphism and inheritance in Java programming.
<b>22CSE36L.3</b>	Demonstrate the concepts of different access control modifiers object-oriented programming and File Processing.
<b>22CSE36L.4</b>	Demonstrate the concepts of exception handling and multithreading and String Handling.
<b>22CSE36L.5</b>	To develop applications using generic programming and event handling.
<b>22CSE36L.6</b>	Develop an application using Java concepts.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Java: The Complete Reference	Herbert Schildt	TATA McGraw-Hill publications	7 <sup>th</sup> Edition, 2009
<b>Reference Books</b>				
1	Programming with Java	E Balagurusamy	McGraw Hill Education	6 <sup>th</sup> Edition, 2019
2	Thinking in Java	Bruce Eckel	Prentice Hall	4 <sup>th</sup> Edition, 2006

**Web links and Video Lectures (e-Resources):**

- **Programming in Java:** -[https://onlinecourses.nptel.ac.in/noc22\\_cs47/preview](https://onlinecourses.nptel.ac.in/noc22_cs47/preview)

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE36L.1	1	1		1	1	2						1	1	1
22CSE36L.2	1		1	1									1	2
22CSE36L.3			1	1			1			1		1		
22CSE36L.4										1	1	1		
22CSE36L.5		1	1						1			1		1
22CSE36L.6	2	1	1	1	1	3	1			1			1	1

1: Low 2: Medium 3: High

<b>Universal Human Values- II</b>			
Course Code	<b>22UHV37</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	<b>02</b>	Exam Hours	02
<p><b>Course Learning Objectives:</b> This introductory course input is intended:</p> <ol style="list-style-type: none"> <li>1. To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.</li> <li>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.</li> <li>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.</li> </ol>			
<b>Module-1 Introduction to Value Education</b>			
<p>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.</p> <p>Activities: Sharing about Oneself, Exploring Human Consciousness and Exploring Natural Acceptance. <span style="float: right;"><b>5 Hours</b></span></p>			
<b>Module-2 – Harmony in the Human Being</b>			
<p>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.</p> <p>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. <span style="float: right;"><b>5 hours</b></span></p>			
<b>Module 3 – Harmony in the Family and Society</b>			
<p>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.</p> <p>Activities: Exploring the Feeling of Trust, Exploring the Feeling of Respect and Exploring the Feeling systems to fulfil Human Goal. <span style="float: right;"><b>5 hours</b></span></p>			
<b>Module-4 – Harmony in the Nature/Existence</b>			
<p>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.</p> <p>Activities: Exploring the Four Orders of Nature and Co-existence in Existence. <span style="float: right;"><b>5 hours</b></span></p>			
<b>Module-5 – Implications of the Holistic Understanding – a Look at Professional Ethics</b>			
<p>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</p> <p>Activities: Exploring Ethical Human Conduct, Humanistic Models in Education and steps of Transition towards Universal Human Order. <span style="float: right;"><b>5 hours</b></span></p>			

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

<b>Sl. No.</b>	<b>Title of the Book</b>	<b>Name of the Author/s</b>	<b>Name of the Publisher</b>	<b>Edition and Year</b>
<b>Textbooks</b>				
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
<b>Reference Books</b>				
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](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](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/1nh9m5ibEtvMyqekeiexAJtfbdNtm6-?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=PLWDeKF97v9SP7wSlapZcQRrT7OH0ZIGC4>

### 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

<b>Biology for Engineers</b>			
Course Code	<b>22BFE37</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	<b>02</b>	Exam Hours	02
<b>Course Learning Objectives:</b>			
<ol style="list-style-type: none"> <li>1. To bring awareness of biological concepts to engineering students</li> <li>2. To introduce the building blocks of life and their complexity</li> <li>3. To encourage interdisciplinary studies and projects</li> <li>4. To appreciate the discoveries that mimic nature and its working</li> <li>5. To inculcate nature-inspired design and operational principles</li> </ol>			
<b>Module-1</b>			<b>5 Hours</b>
<b>Basic Cell Biology:</b> 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.			
<b>Module-2</b>			<b>5 Hours</b>
<b>Biochemistry and Molecular Aspects of Life:</b> 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.			
<b>Module-3</b>			<b>5 Hours</b>
<b>Bioinspired Engineering based on human physiology:</b> 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).			
<b>Module-4</b>			<b>5 Hours</b>
<b>Relevance of Biology as an interdisciplinary approach:</b> 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).			
<b>Module-5</b>			<b>5 Hours</b>
<b>Bioinspired Algorithms and Applications:</b> 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.			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22BFE37.1</b>	Discuss how the cell forms the basic building block of life
<b>22BFE37.2</b>	Distinguish between transcription and translation
<b>22BFE37.3</b>	Describe the role played by proteins within the cell
<b>22BFE37.4</b>	Analyze the role of bioinspired design in novel applications
<b>22BFE37.5</b>	Apply bioinspired design principles to other domains
<b>22BFE37.6</b>	Implement a simple genetic algorithm



Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Text Books</b>				
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	5 <sup>th</sup> Edition, 2012
<b>Reference books</b>				
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	1 <sup>st</sup> Edition, 2016
4	A Practical Guide to Bio-inspired Design	Hashemi Farzaneh, Helena, Lindemann, Udo	Springer	2019

#### Web links/Video Lectures/MOOCs

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

<b>IOT ENABLED PROTOTYPING</b>			
Course Code:	<b>22IEP38</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	<b>01</b>	Exam Hours	02
<b>Course Learning Objectives:</b>			
<ol style="list-style-type: none"> <li>1. Understand the IoT concepts such as sensing, actuation, and communication.</li> <li>2. Development of Internet of Things (IoT) prototypes—including devices for sensing, actuation, processing, and communication and Protocols</li> <li>3. Understand the significance of Project Management and the different techniques of planning</li> <li>4. To introduce fundamental aspects of intellectual property rights, Govt. policies on IPR, and patentability search techniques.</li> </ol>			
<b>Module 1</b>			
<b>Internet of Things – Hardware / System Design</b>			
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.			
<b>6 Hours</b>			
<b>Module 2</b>			
<b>Internet of Things</b>			
<b>Networking in IoT:</b>			
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)			
<b>IoT in Web/ Cloud Platform:</b>			
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.			
<b>6 Hours</b>			
<b>Module 3</b>			
<b>Project Planning and Management</b>			
Project initiation, Project charter, Project planning, and implementation, Scheduling and costing, Project monitoring and control, Project closure and reports.			
<b>6 Hours</b>			
<b>Module 4</b>			
<b>Intellectual Property Rights</b>			
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.			
<b>6 Hours</b>			
<b>Course Project</b>			
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.			
<b>6 Hours</b>			
<b>Course Outcomes:</b> At the end of the course, the student will be able to:			
<b>22IEP38.1</b>	Analyze the basics of IoT and protocols.		
<b>22IEP38.2</b>	Develop IoT-based prototypes to solve industrial and societal problems.		
<b>22IEP38.3</b>	Apply appropriate approaches to plan a new project and develop a project schedule.		

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

<b>Sl. No.</b>	<b>Title of the Book</b>	<b>Name of the Author/s</b>	<b>Name of the Publisher</b>	<b>Edition and Year</b>
<b>Reference Books</b>				
1	Internet of Things (A Hands-on-Approach)	Vijay Madiseti and Arshdeep Bahga	Orient Blackswan Private Limited	1 <sup>st</sup> 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)	4 <sup>th</sup> 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
<b>22IEP38.1</b>			2		2				2	2				
<b>22IEP38.2</b>			2								3			
<b>22IEP38.3</b>					2						2			
<b>22IEP38.4</b>								1		2				
<b>22IEP38.5</b>								1	2	2				

1: Low 2: Medium 3: High

<b>Industry Oriented Training - Business Etiquettes</b>			
Course Code	<b>22ITB39A</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02
<b>Course Learning Objectives:</b>			
6. Know the components of self-introduction 7. Develop a resume with the inclusion of core competencies 8. Involve and contribute to group discussions 9. Develop effective communication to succeed in the professional career 10. Know the etiquettes of digital communication			
<b>Module-1</b>			
<b>Self-Introduction &amp; Essentials of grooming</b>			
<b>Self-Introduction:</b> Learn the secret to introducing Yourself, Things to avoid when introducing yourself. Activity: Video record the self-introduction. <b>Essentials of grooming:</b> Creating the first impression, what does the well-dressed man wear? What does the well-dressed woman wear? Personal hygiene and habits. <b>4 Hours</b>			
<b>Module-2</b>			
<b>Resume Writing</b>			
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. <b>4 Hours</b>			
<b>Module-3</b>			
<b>Group Discussion</b>			
Types, process, Evaluation criteria, Do's and Don'ts Activity: Group discussions have to be held during the training sessions. <b>4 Hours</b>			
<b>Module-4</b>			
<b>Communicate effectively</b>			
Build a Story, Just a Minute, Group Activities, Team building activities, Role Play, Presentation Skills. <b>4 Hours</b>			
<b>Module-5</b>			
<b>Digital right and wrong</b>			
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. <b>4 Hours</b>			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22ITB39A.1</b>	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
<b>22ITB39A.2</b>	Develop a resume inclusive of core competencies, and action verbs which are compatible with Applicant Tracking Systems
<b>22ITB39A.3</b>	Demonstrate the types, process and evaluation process of Group Discussion and carry out effective group discussions
<b>22ITB39A.4</b>	Develop skills required for effective communication
<b>22ITB39A.5</b>	Associate and be accustomed to the etiquette to be followed during online meetings

<b>Sources</b>	
1.	English for Common Interactions in the Workplace: Basic Level: Coursera: <a href="https://www.coursera.org/learn/english-common-interactions-workplace-basic-level">https://www.coursera.org/learn/english-common-interactions-workplace-basic-level</a>
2.	Personal Communication-Introduce Yourself With Confidence: <a href="https://www.udemy.com/course/how-to-introduce-yourself/">https://www.udemy.com/course/how-to-introduce-yourself/</a>
3.	Professionalism, Grooming and Etiquette: <a href="https://www.edx.org/course/professionalism-grooming-and-etiquette">https://www.edx.org/course/professionalism-grooming-and-etiquette</a>
4.	How to Write a Resume: <a href="https://www.coursera.org/learn/how-to-write-a-resume#syllabus">https://www.coursera.org/learn/how-to-write-a-resume#syllabus</a>
5.	Group Discussion Strategies: <a href="https://www.udemy.com/course/group-discussion-strategies/">https://www.udemy.com/course/group-discussion-strategies/</a>
6.	Communication Strategies for a Virtual Age: <a href="https://www.coursera.org/learn/communication-strategies-virtual-age#syllabus">https://www.coursera.org/learn/communication-strategies-virtual-age#syllabus</a>
<b>References</b>	
1.	<a href="https://simplifytraining.com/course/personal-hygiene-and-good-grooming/">https://simplifytraining.com/course/personal-hygiene-and-good-grooming/</a>
2.	<a href="https://www.udemy.com/course/group-discussion-strategies/">https://www.udemy.com/course/group-discussion-strategies/</a>
3.	<a href="https://www.educba.com/course/group-discussion/">https://www.educba.com/course/group-discussion/</a>
4.	<a href="https://getrafiki.ai/meetings/rules-of-virtual-meeting-etiquette-every-sales-professional-should-follow/">https://getrafiki.ai/meetings/rules-of-virtual-meeting-etiquette-every-sales-professional-should-follow/</a>
5.	<a href="https://thedigitalworkplace.com/articles/online-meeting-etiquette-for-attendees/">https://thedigitalworkplace.com/articles/online-meeting-etiquette-for-attendees/</a>
6.	<a href="https://rigorousthemes.com/blog/virtual-meeting-etiquette-guidelines-ground-rules/">https://rigorousthemes.com/blog/virtual-meeting-etiquette-guidelines-ground-rules/</a>

### Course Articulation Matrix

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

1: Low 2: Medium 3: High

<b>Industry Oriented Training - Computing Skills</b>			
Course Code	<b>22ITC39B</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02
<b>Course Learning Objectives:</b>			
<ol style="list-style-type: none"> <li>1. Use logical conditions for problem-solving and also introduce the concepts of arrays</li> <li>2. Know functions, function calls, and parameter passing</li> <li>3. Introduce algorithms and appreciate their importance in problem-solving</li> <li>4. Introduce the core concepts of OOP's</li> <li>5. Differentiate between front-end &amp; back-end development and recognize the use of database management</li> </ol>			
<b>Module-1</b>			
<b>Introduction to computing constructs</b>			
<p>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.</p> <p>Arrays &amp; strings: Create arrays of characters (strings), use the null terminator, and manipulate strings.</p>			
<b>4 Hours</b>			
<b>Module-2</b>			
<b>Functions &amp; Pointers</b>			
<p>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.</p>			
<b>4 Hours</b>			
<b>Module-3</b>			
<b>Algorithm analysis</b>			
<p>Introduction to Algorithm Analysis, Big-O, Big-O Examples, Dynamic Array Operations, Bubble Sort, Selection Sort, Insertion Sort, Recursion, Recursive Binary Search, Merge Sort.</p>			
<b>4 Hours</b>			
<b>Module-4</b>			
<b>Object-oriented programming</b>			
<p>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.</p>			
<b>4 Hours</b>			
<b>Module-5</b>			
<b>Frontend and backend development</b>			
<p>UI, Database management: DBMS overview, Relational Data Model and the CREATE TABLE Statement, Basic Query Formulation with SQL.</p>			
<b>4 Hours</b>			

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

1: Low 2: Medium 3: High

# **IV Semester**



<b>Discrete Mathematical Structures</b>			
Course Code	<b>22CSE41</b>	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
<b>Course Learning Objectives:</b>			
<ul style="list-style-type: none"> <li>• Use propositional logic in knowledge representation.</li> <li>• Apply principles of counting in computer applications</li> <li>• Find relation between different sets</li> <li>• Apply recurrence relations in real life problems</li> <li>• Use graph theory in computer applications</li> </ul>			
<b>Module-1 (8 hours)</b>			
<b>Fundamentals of Logic:</b> Basic Connectives and Truth Tables, Logical Equivalence: The laws of logic, Rules of inference. Open Statement, Quantifiers			
<b>Module-2 (8 hours)</b>			
<b>Fundamental Principles of Counting:</b> Method of mathematical induction, The Rule of Sum and Product, Permutations, Combinations, The Binomial Theorem			
<b>Module-3 (8 hours)</b>			
<b>Relations and Functions:</b> Cartesian Products and Relations, Properties of relations, Equivalence Relations and Partitions. Functions, Function composition and Inverse function			
<b>Module-4 (8 hours)</b>			
<b>Enumeration and Generating Functions:</b> Inclusion-exclusion principle, rook polynomials. First order linear recurrence relation, Second order linear homogeneous recurrence relations with constant coefficients			
<b>Module-5 (8 hours)</b>			
<b>Graph Theory:</b> Graphs and sub graphs, Graph Isomorphism, Vertex degree, Planar Graphs, Graph Coloring, Trees and Sorting, and Prefix Codes.			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE41.1</b>	Apply knowledge of propositional logic in truth verification
<b>22CSE41.2</b>	Demonstrate the use of principles of counting in computer science
<b>22CSE41.3</b>	Recognize relations in real life applications
<b>22CSE41.4</b>	Find applications of equivalence relation
<b>22CSE41.5</b>	Apply inclusion principle, rook polynomial in computer applications
<b>22CSE41.6</b>	Use graph theory in computer science

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Discrete and Combinatorial Mathematics	Ralph P. Grimaldi and B V Ramana	Pearson Education, Asia,	5 <sup>th</sup> Edition, 2017
2	A first look at Graph Theory	John Clark and D. A. Holton	World Scientific Publishers	2 <sup>nd</sup> Edition, 2011

Reference Books				
1	Discrete Mathematical Structures with Applications to Computer Science	J.P. Tremblay and R. Manohar	Tata – McGraw Hill Publications	1 <sup>st</sup> Edition, 2017
2	Discrete Mathematics and its Applications	Kenneth H. Rosen	Tata – McGraw Hill Publications	7 <sup>th</sup> Edition, 2017
3	Topics in Algebra	I N Herstein	Wiley Eastern Limited	2 <sup>nd</sup> Edition, 1993

**Web links and Video Lectures (e-Resources):**

- <https://youtu.be/yGC1weQ1n2o>
- <https://youtu.be/xaxt-zNIRRG>
- <https://youtu.be/7mhvA5L7KqY>
- <https://youtu.be/eHwjcRM6UhA>
- <https://youtu.be/cqSZnON00OQ>
- <https://youtu.be/J7g5HMZgKMU>

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE41.1	2		1											
22CSE41.2		2										1		
22CSE41.3	2	1												
22CSE41.4		2										1		
22CSE41.5	2											1		
22CSE41.6	1	2												

1: Low 2: Medium 3: High

<b>Design and Analysis of Algorithms</b>			
Course Code	<b>22CSE42</b>	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
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• Apply appropriate methods to solve a given problem and validate its correctness.</li> <li>• Ability to analyze time complexity of the algorithms.</li> <li>• Implementation of various algorithmic techniques like Greedy strategy, Divide and Conquer approach, Dynamic Programming and Backtracking.</li> <li>• Synthesize efficient algorithms in common engineering design situations.</li> <li>• Understand the limitations of the algorithmic power</li> </ul>			
<b>Module-1: Introduction and Analysis of Algorithms (8 Hours)</b>			
<p><b>Introduction:</b> What is an Algorithm? Fundamentals of Algorithmic Problem Solving, Important Problem Types. <b>Analysis:</b> Analysis Framework, Asymptotic Notations and Basic Efficiency classes, Mathematical analysis of Non-Recursive and Recursive Algorithms with Examples. Empirical Analysis of Algorithms.  <b>Chapter 1: 1.1-1.3, Chapter 2: 2.1-2.4, 2.6</b></p>			
<b>Module-2: Divide and Conquer (8 Hours)</b>			
<p><b>Divide and Conquer:</b> General method, Recurrence equation, Master Theorem, Merge sort, Quick sort, Strassen's matrix multiplication. <b>Decrease and Conquer:</b> Binary search. <b>Transform and Conquer:</b> AVL Trees, Heaps and Heap sort.  <b>Chapter 5: 5.1,5.2,5.4 Chapter 4: 4.4, Chapter 6: 6.3, 6.4</b></p>			
<b>Module-3: Greedy method (8 hours)</b>			
<p><b>Greedy method:</b> General method. <b>Minimum cost spanning trees:</b> Prim's Algorithm, Kruskal's Algorithm. <b>Single source shortest paths:</b> Dijkstra's Algorithm. <b>Optimal Tree Problem:</b> Huffman Trees and Codes. <b>Space and Time Tradeoffs:</b> Sorting by Counting, B-Trees  <b>Chapter 9: 9.1-9.4, Chapter 7: 7.1, 7.4</b></p>			
<b>Module-4: Dynamic programming (8 hours)</b>			
<p><b>Dynamic programming:</b> Knapsack problem with memory functions, Optimal Binary Search Trees, Transitive Closure-Warshall's Algorithm, and All Pairs Shortest Paths-Floyd's Algorithm. <b>Limitations of Algorithm Power:</b> P, NP and NP- Complete Problems.  <b>Chapter 8: 8.2-8.4, Chapter 11: 11.3</b></p>			
<b>Module-5: Backtracking (8 hours)</b>			
<p><b>Backtracking:</b> N-Queens problem, Hamiltonian circuit Problem, Sum of subsets problem. <b>Branch and Bound:</b> Assignment problem, Knapsack problem, Travelling Sales Person problem  <b>Chapter 12: 12.1, 12.2</b></p>			

<b>PRACTICAL MODULE</b>
Design, develop, and implement the specified algorithms for the following problems using Java language under LINUX /Windows environment. Netbeans / Eclipse IDE tool can be used for development and demonstration.

**Regular Experiments (All experiments to be executed in laboratory)**

1. Merge Sort: To sort n randomly generated integers and perform empirical analysis.
2. Quick Sort: To sort n randomly generated integers and perform empirical analysis.
3. Binary Search: To search a key in the list of n integers
4. Prim's algorithm: To find the Minimum Spanning Tree of an undirected graph.
5. Kruskal's Algorithm: To find the Minimum Spanning Tree of an undirected graph.
6. Use dynamic programming to
  - i. find all pairs shortest distance between nodes of a graph using Floyd's Algorithm and
  - ii. determine the transitive closure of a directed graph using Warshalls Algorithm
7. Knapsack Problem: To solve 0/1 Knapsack problem using dynamic programming
8. Subset problem: To solve the sum of subset problem using branch and bound method.

**Open Ended Experiments(Any one/two)**

9. Heap Sort: Sort array of n numbers using Heap Sort
10. N-Queens Problem: Generate all solutions of N-Queens problem.
11. Dijkstra's algorithm: From a given vertex in a weighted connected graph, find shortest paths to other vertices.
12. Any other experiment covering the concepts of algorithms.

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

<b>22CSE42.1</b>	Interpret the time and space complexity of algorithms which provides solutions to the given problem
<b>22CSE42.2</b>	Identify the problems from the set that can be solved using divide and conquer techniques and apply the technique to obtain the solutions.
<b>22CSE42.3</b>	Apply the technique of greedy algorithms in real life applications to get the optimal solution
<b>22CSE42.4</b>	Apply the dynamic programming design technique to solve various problems
<b>22CSE42.5</b>	Differentiate the problems that can be solved using backtracking method and other general design techniques for given set of problems
<b>22CSE42.6</b>	Analyze the limitations of algorithm power

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Introduction to the Design and Analysis of Algorithms	Anany Levitin	Pearson	3 <sup>rd</sup> Edition, 2023
<b>Reference Books</b>				
1	Introduction to Algorithms	Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein	Prentice Hall India	3 <sup>rd</sup> Edition, 2014
2	Computer Algorithms	Ellis Horowitz, Satraj Sahni and Rajasekaran	Galgotia Publications P Ltd	2013

**Web links/Video Lectures/MOOCs/papers**

- NPTEL Design and Analysis of Algorithms by Prof. Madhavan Mukund: <https://nptel.ac.in/courses/106106131>
- NPTEL Fundamental Algorithms: Design and Analysis by Prof. Sourav Mukhopadhyay: [https://onlinecourses.nptel.ac.in/noc22\\_cs01/preview](https://onlinecourses.nptel.ac.in/noc22_cs01/preview)
- GeekforGeeks, Algorithms: <https://www.geeksforgeeks.org/fundamentals-of-algorithms/>
- Tutorialspoint, Design and Analysis of Algorithms Tutorial: [https://www.tutorialspoint.com/design\\_and\\_analysis\\_of\\_algorithms/index.html](https://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.html)

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE42.1		2		3										3
22CSE42.2		3	3										2	
22CSE42.3			3										2	
22CSE42.4			3										2	
22CSE42.5				3									2	
22CSE42.6				3										3

**1: Low 2: Medium 3: High**

<b>Database Management System</b>			
Course Code	<b>22CSE43</b>	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
<b>Course Learning Objectives:</b>			
<ul style="list-style-type: none"> <li>• Provide a strong foundation in database concepts, technology, and practice.</li> <li>• Practice SQL programming through a variety of database problems.</li> <li>• Demonstrate the use of concurrency and transactions in the database.</li> <li>• Design and build database applications for real world problems.</li> <li>• Develop applications to interact with databases.</li> </ul>			
<b>Module-1 Databases and Database Concept and Model (8 hours)</b>			
<b>Introduction to Databases:</b> Introduction, Simplified database system environment, Characteristics of database approach, Actors on the scene, Workers behind the scene, Advantages of using the DBMS approach.			
<b>Database Concepts and Architectures:</b> Data Models, Schemas and Instances. Three schema architecture and data independence, database languages and interfaces, Component modules of a DBMS and their Interactions.			
<b>Data Model:</b> 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.			
<b>Chapter No: 1.1, 1.3 to 1.6, 2.1 to 2.4.1, 3.1, 3.3.1, 3.3.2, 3.4, 3.7</b>			
<b>Module-2 Relational Model and SQL (8 hours)</b>			
<b>Relational Model:</b> Relational Model Concepts, Relational Model Constraints and schemas, Update Operations and Dealing with Constraint violations.			
<b>SQL:</b> 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.			
<b>Chapter No: 5.1 to 5.3, 6.1 to 6.4, 7.1 to 7.4</b>			
<b>Module-3 Relational Algebra and Database Application Development (8 hours)</b>			
<b>Relational Algebra &amp; Design:</b> Unary and Binary relational operations, Relational Algebra Operations, Additional Relational Operations, Examples of Queries in Relational Algebra.			
<b>Database Application Development:</b> Embedded SQL, Dynamic SQL, SQLJ, Database Programming with Function calls: SQL and JDBC, Database Stored Procedures.			
<b>Chapter No. 8.1 to 8.5, 10.2 to 10.4</b>			
<b>Module-4 Normalization (8 hours)</b>			
<b>Normalization and its Algorithms:</b> 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. <b>Chapter No: 14.1 to 14.7, 15.2 to 15.4</b>			
<b>Module-5 (8 hours)</b>			
<b>Transaction Processing:</b> 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. 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.

2. Write ER Diagram Bank Database, College Database. Demonstrate mapping of ER-diagram to Relational Schema model.

3. The Company database of an organization has a table with following attributes

**EMPLOYEE(empid:int, emp\_name:string, DOJ:date, department:string, salary:decimal)**

i. Create the above table using suitable constraints.

ii. Enter the five tuples into the table

iii. Display all the number of employees working in each department.

iv. Find the sum of the salaries of all employees.

v. Find the sum and average of the salaries of employees of a particular department.

vi. Find the emp\_id and emp\_name of all the Employees who has a Salary not equal to 25000 or 30000.

vii. Display all records who's name starts and ends with A and has at least 4 character

4. The College Database has following tables

**DEPARTMENT(DEPT\_ID: int, DEPT\_NAME: string)**

**STUDENT(STUD\_ID:int, STUD\_NAME:String,AGE: int, DOB: date, ADDRESS: String,DEPT\_NO:int)**

**SUBJECT(SUBJECT\_ID:int, SUBJECT\_NAME:String,DEPT\_NO:int)**

**TEACHER(TEACHER\_ID:int,TEACHER\_NAME:String,SUB\_NO:int, DEPT\_NO: int)**

Create tables using suitable constraints (PRIMARY KEY, NOT NULL, FOREIGN KEY), Insert 10 Sample data into the table and perform following operations

i. List all the students' details studying in Computer Science Department

ii. List all the teachers teaching in EEE Department

iii. List the subject offered by Computer Science Department

iv. Change the date type of STUD\_ID from integer to string

v. Add new attribute STUD\_Phno to the existing STUDENT table

vi. Combine each row of Student table with each row of Subject table

vii. Create a student table view

viii. Display the names of the students who opted DBMS subject

5. Consider the following database of student's enrolment in courses and books adopted for each course.

**STUDENT (regno: string, name: string, major: string, bdate: date)**

**COURSE (course-no: int, cname: string, dept: string)**

**ENROLL (reg-no: string, course-no: int, sem: int, marks: int)**

**BOOK\_ADOPTION (course\_no:int, Sem:int,book\_isbn:int)**

**TEXT (book-isbn: int book-title: string, publisher: string, author: string)**

- i. Create the above tables by properly specifying the primary keys and the foreign keys
- ii. Enter at least five tuples for each relation.
- iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
- iv. Produce a list of text books (include Course-no, book-isbn, book-title) in the alphabetical order for courses offered by the 'Compute Science' department that use more than two books.
- v. List any department that has all its adopted books published by a specific publisher.
- vi. Delete the view created
- vii. Display all records whose name starts with John

6. Consider the following database

**SALESMAN(Salesman\_id:int, Salesman\_Name:string, City:string,Comission:int)**

**ORDERS(ord\_id:int, ord\_date:date, Purch\_amt:int, Customer\_id:int, Salesman\_id:int)**

**CUSTOMERS(Customer\_id:int, Customer\_name:string, City:stringGrade:int, Salesman\_id:int)**

- i. Create the above tables by properly specifying the primary keys and the foreign keys
- ii. Enter atleast 10 tuples for each relation.
- iii. Find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city
- iv. Find those orders where the order amount exists between 500 and 2000. Return ord\_no, purch\_amt, cust\_name, city.
- v. List the salesmen who either work for one or more customers or yet to join any of the customer. The customer, may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier

7. Consider the following database

**Sailors(sid:int, Sname:string, Rating:int, age:real)**

**Boats (bid: integer, bname:String, color:String)**

**Reserves (Sid:int, Bid:int, day:date)**

- i. Create the above tables by properly specifying the primary keys and the foreign keys
- ii. Enter at least 10 tuples for each relation.
- iii. Find the ages of sailors whose name begins and ends with B and has at least ~~two~~ three characters.
- iv. Find the names of sailors who have reserved a red or a green boat.
- v. Find the sailors with the highest rating
- vi. Find the names of sailors who have reserved all boats.



<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE43.1</b>	Identify the various elements of Database Management Systems and to draw an E-R diagram.
<b>22CSE43.2</b>	Solve a given problem statement, analyze the entities, its types and their relations.
<b>22CSE43.3</b>	Take part in writing queries using Relational Algebra, SQL and PL/SQL.
<b>22CSE43.4</b>	Examine the normalizations for the development of application software.
<b>22CSE43.5</b>	Determine the concepts of transaction, concurrency control, recovery and security in the database.
<b>22CSE43.6</b>	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
<b>Textbooks</b>				
1	Fundamentals of Database Systems	Ramez Elmasri and Shamkant B. Navathe,	Pearson	7 <sup>th</sup> Edition, 2017
<b>Reference Books</b>				
1	Database System Concepts	Abraham Silberschatz, Henry F. Korth, S. Sudharshan, Tata	McGraw Hill,	6 <sup>th</sup> Edition, 2011
2	Database management systems	Ramakrishnan, and Gehrke,	McGraw Hill	3 <sup>rd</sup> Edition, 2014

<b>Web links/Video Lectures/MOOCs/papers</b>	
<ul style="list-style-type: none"> <li>• <a href="https://www.tutorialspoint.com/dbms/">https://www.tutorialspoint.com/dbms/</a></li> <li>• <a href="https://www.w3schools.com/sql/">https://www.w3schools.com/sql/</a></li> <li>• <a href="https://www.codecademy.com/learn/learn-sql">https://www.codecademy.com/learn/learn-sql</a></li> <li>• <a href="https://in.udacity.com/">https://in.udacity.com/</a></li> <li>• <a href="https://www.geeksforgeeks.org/dbms/">https://www.geeksforgeeks.org/dbms/</a></li> </ul>	

### Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
<b>22CSE43.1</b>	2	2			1									1
<b>22CSE43.2</b>	2					2								
<b>22CSE43.3</b>		2	2											1
<b>22CSE43.4</b>						1								1
<b>22CSE43.5</b>			2				1						1	1
<b>22CSE43.6</b>						1			1				1	

1: Low 2: Medium 3: High

<b>Operating System</b>			
Course Code	<b>22CSE44</b>	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	Credits	03
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• Demonstrate need for Operating system, types and services.</li> <li>• Apply suitable techniques for management of various resources.</li> <li>• Use Processor, Memory, Storage and File system commands.</li> <li>• Demonstrates the use of Memory and Virtual memory management.</li> <li>• Analyze the concept of Deadlock and Process synchronization.</li> </ul>			
<b>Module-1 Introduction to operating systems (8 hours)</b>			
<p><b>Introduction to operating systems:</b> What Operating Systems Do: User View, System View, Computer-System Operation: Computer-System Operation, Storage Structure, I/O Structure, Computer-System Architecture: Single-Processor Systems, Multiprocessor Systems, Clustered Systems, Operating System Structure: Multiprogramming, Timesharing, Operating-System Operations: Dual-Mode and Multimode Operation, Timer, Protection and Security, Computing Environments: Traditional Computing, Mobile Computing, Distributed Systems, Client–Server Computing, Peer-to-Peer Computing, Virtualization, Cloud Computing, Real-Time Embedded Systems.</p> <p><b>Chapters: 1 (1.1 – 1.5, 1.9, 1.11)</b></p>			
<b>Module-2 , Operating System Services and Process (8 hours)</b>			
<p><b>Operating System Services:</b> Operating-System Services, System Calls, Types of System Calls: Process Control, File Management, Device Management, Information Maintenance, Communication, Protection, System Programs, Operating-System Structure: Simple Structure, Layered Approach, Microkernels, Modules, Hybrid Systems.</p> <p><b>Process:</b> Process concept: The Process, Process State, Process Control Block, Threads, Process Scheduling: Scheduling Queues, Schedulers, Context Switch</p> <p><b>Chapters: 2 (2.1, 2.3 – 2.5, 2,7) and 3 ( 3.1 – 3.2)</b></p>			
<b>Module-3 Process Operations and Synchronization (8 hours)</b>			
<p><b>Process:</b> Operations on Processes: Process Creation, Process Termination, Interprocess Communication: Shared-Memory Systems, Message-Passing Systems.</p> <p><b>Process Synchronization:</b> The Critical-Section Problem, Peterson’s Solution, Synchronization Hardware, Mutex Locks, Semaphores: Semaphore Usage, Semaphore Implementation, Deadlocks and Starvation, Priority Inversion, Classic Problems of Synchronization: The Bounded-Buffer Problem, The Readers–Writers Problem, The Dining-Philosophers Problem</p> <p><b>Chapters: 3 (3.3 – 3.4) and 5 ( 5.2 – 5.7)</b></p>			
<b>Module-4 CPU Scheduling and Deadlocks (8 hours)</b>			
<p><b>CPU Scheduling:</b> Basic Concepts: CPU–I/O Burst Cycle, CPU Scheduler, Dispatcher, Scheduling Criteria, Scheduling Algorithms: First-Come, First-Served Scheduling, Shortest-Job-First Scheduling, Priority Scheduling, Round-Robin Scheduling, Multilevel Queue Scheduling, Multilevel Feedback Queue Scheduling</p> <p><b>Deadlocks :</b> System Model, Deadlock Characterization: Necessary Conditions, Resource-Allocation Graph, Methods for Handling Deadlocks, Deadlock Prevention: Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait, Deadlock Avoidance: Safe State, Resource-Allocation-Graph Algorithm, Banker’s Algorithm, Deadlock Detection: Single Instance of Each Resource Type, Several Instances of a Resource Type, Detection-Algorithm Usage, Recovery from Deadlock: Process Termination, Resource Preemption.</p> <p><b>Chapters: 6 (6.1 – 6.3) and 7 (7.1 – 7.7)</b></p>			
<b>Module-5 Main Memory and Virtual Memory (8 hours)</b>			
<p><b>Main Memory :</b> Swapping: Standard Swapping, Swapping on Mobile Systems, Contiguous Memory Allocation: Memory Protection, Memory Allocation, Fragmentation, Segmentation: Basic Method, Segmentation Hardware, Paging: Basic Method, Hardware Support</p> <p><b>Virtual Memory:</b> Demand Paging: Basic Concepts, Performance of Demand Paging, Copy-on-Write, Page Replacement: Basic Page Replacement, FIFO Page Replacement, Optimal Page Replacement, LRU Page</p>			

Replacement  
**Chapters: 8 (8.2 – 8.5) and 9 (9.2 – 9.4)**

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE44.1</b>	Ability to analyze the design of Operating System operations and Services.
<b>22CSE44.2</b>	Demonstrates Process Management and Multi-threaded programming.
<b>22CSE44.3</b>	Illustrate the mechanism of Process Synchronization and Deadlock.
<b>22CSE44.4</b>	Illustrate Memory and Virtual Memory Management.
<b>22CSE44.5</b>	Implementation of File System and Space allocation method.
<b>22CSE44.6</b>	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
<b>Textbooks</b>				
1	Operating System Concepts	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne	Wiley-India	9th Edition, 2012
<b>Reference Books</b>				
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

**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
<b>22CSE44.1</b>	-	-	-	-	-	-	-	-	-	-	-	-	-	2
<b>22CSE44.2</b>	-	2	-	-	-	-	-	-	-	-	-	-	-	-
<b>22CSE44.3</b>	-	2	-	-	-	-	-	-	-	-	-	-	-	-
<b>22CSE44.4</b>	-	-	-	-	-	2	-	-	-	-	-	-	-	2
<b>22CSE44.5</b>	-	-	-	-	-	2	-	-	-	-	-	-	-	2
<b>22CSE44.6</b>	-	-	2	-	-	-	-	-	-	-	-	-	-	-

1: Low 2: Medium 3: High

<b>Data Analysis using R Programming</b>			
Course Code	<b>22CSE451</b>	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	Credits	03
<b>Course Learning Objectives:</b>			
<ul style="list-style-type: none"> <li>• Build R scripts for importing data and manage subsets of data.</li> <li>• Apply R based functions to manage data tables and plot the data.</li> <li>• Analyze the data using R scripts by applying statistical functions.</li> <li>• Demonstrate the visualization of data using R functions.</li> </ul>			
<b>Module-1 First steps in R (8 hours)</b>			
<b>First steps in R:</b> Typing in small datasets, Concatenating Data with the c Function, Combining Variables with the c, cbind, and rbind Functions, Combining Data with the vector Function, Combining Data Using a Matrix, Combining Data with the data. frame Function, Combining Data Using the list Function, <b>Importing Data:</b> Importing Excel Data, Accessing Data from Other Statistical Packages, Accessing a Database <b>Textbook 1: Chapter 2.1.1,2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 2.1.7, 2.2, 2.2.1, 2.2.2, 2.2.3</b>			
<b>Module-2 Accessing Variables and Managing Subsets of Data (8 hours)</b>			
<b>Accessing Variables and Managing Subsets of Data:</b> Accessing Variables from a Data Frame, Accessing Subsets of Data, Combining Two Datasets with a Common Identifier, Exporting Data, Recoding Categorical Variables <b>Textbook 1: Chapter 3.1, 3.2, 3.3, 3.4, 3.5</b>			
<b>Module-3 Functions (8 hours)</b>			
<b>Simple Functions:</b> The tapply Function, The sapply and lapply Functions, The summary Function, The table Function, <b>An Introduction to Basic Plotting Tools:</b> The plot Function, Symbols, Colours, and Sizes, Adding a Smoothing Line <b>Textbook 1: Chapter 4.1, 4.2, 4.3, 4.4, 5.1, 5.2, 5.3</b>			
<b>Module-4 Statistics with R (8 hours)</b>			
<b>Basic Mathematics:</b> Basic mathematical functions, matrix operations, numerical integration, differentiation, optimization. <b>Descriptive Statistics using R:</b> Structuring variables according to Type, Data Tables, Numerical summaries <b>Textbook 2: Chapter 10.1, 10.2, 10.3, 10.4, 10.5, 11.2, 11.3, 11.4</b>			
<b>Module-5 Graphical Representations with R (8 hours)</b>			
<b>Graphical Representations:</b> Plotting qualitative variables, Plotting ordinal variables, Plotting discrete quantitative variables, Plotting continuous quantitative variables, Graphical representations in a bi-variate setting <b>Textbook 2: Chapter 11.6.1, 11.6.2, 11.6.3, 11.6.4, 11.6.5</b>			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE451.1</b>	Demonstrate data manipulation using R programming.
<b>22CSE451.2</b>	Utilize R scripts to access variables and manage subsets of data.
<b>22CSE451.3</b>	Inspect the data using R functions and plotting tools.
<b>22CSE451.4</b>	Solve statistical problems for the given data using R.
<b>22CSE451.5</b>	Develop solutions for visualizing the data using R programming.
<b>22CSE451.6</b>	Apply statistical analysis and provide visual representations of the data.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	A Beginner's Guide to R	Alain F. Zuur, Elena N. Ieno, Erik H.W.G. Meesters	Springer	1 <sup>st</sup> Edition, 2009
2	The R Software-Fundamentals of Programming and Statistical Analysis	Pierre Lafaye de Micheaux, Rémy Drouilhet, Benoit Lique	Springer	1 <sup>st</sup> Edition, 2013
<b>Reference Books</b>				
1	Introduction to Statistics and Data Analysis	Christian Heumann, Michael Schomaker and Shalabh	Springer	1 <sup>st</sup> Edition, 2016

**Web links and Video Lectures (e-Resources):**

- R project website: <https://www.r-project.org>
- Sample R projects: <https://github.com/veeralakrishna/Datacamp-Project-Solutions-R>
- Infosys Springboard: <https://infosyspringboard.onwingspan.com/web/en/login>

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
22CSE451.1					3									3
22CSE451.2				3	2									3
22CSE451.3					3									3
22CSE451.4				2										3
22CSE451.5				3										3
22CSE451.6					2					3				3

1: Low 2: Medium 3: High

<b>SOFTWARE TESTING</b>			
Course Code	<b>22CSE452</b>	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	Credits	03
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>● Understand the Software Testing objectives and the Life cycle.</li> <li>● Differentiate the various testing techniques.</li> <li>● Analyze the problem and derive suitable test cases.</li> <li>● Apply suitable technique for designing of flow graph.</li> <li>● Explain the need for planning and monitoring a process.</li> <li>● To apply the automated testing tools and metrics.</li> </ul>			
<b>Module-1 Introduction to Software Testing (8 hours)</b>			
Context of Testing, Goals of Software Testing, Software Testing Definitions, Model for Software Testing, Software Testing Terminology-Definitions: Failure, Fault/Defect/Bug, Error, Test Case, Test ware, Incident, Test Oracle, Life cycle of Bugs, State of Bugs, Software Testing Life Cycle(STLC), Testing Life Cycle Model, Quality, Quality Assurance and Quality Control. (Text Book1-1.1, 2.2); (Text Book2-1.4,1.6,1.7,2.1.1,2.1.2,2.1.3, 2.2, 2.3.4)			
<b>Module-2 Methods of Testing (8 hours)</b>			
Software testing Terminology and Methodology: Testing Tactics, White Box Testing: What is white box testing?, Static Testing, Structural Testing: Unit/code functional testing, Code coverage Testing, Code complexity Texting, Black box Testing: What is Black box testing?, Why Black box Testing? , When to do a Black box testing?, How to do a Black box Testing?:Requirements based Testing, Positive and negative testing, Boundary value analysis, Decision tables, Equivalence partitioning, State based testing, Compatibility testing, User documentation testing, Domain testing (Text Book1- 3.1, 3.2, 3.3.1-3.3.3, 4.1, 4.2, 4.3, 4.4.1-4.4.9); (Text Book2-2.3.6)			
<b>Module-3 Types and levels of Testing (8 hours)</b>			
Unit Validation Testing, Integration Testing: What is Integration Testing? Integration testing as a type of Testing: Top-down Integration, Bottom-up Integration, Bi-directional Integration, System integration, Choosing integration method, System and Acceptance Testing: Why is system testing done? Categories of System Tests, Functional System Testing, Nonfunctional Testing, Acceptance testing: Alpha testing, Beta testing. (Text Book1-5.1, 5.2.1-5.2.5, 6.2, 6.4, 6.5); (Text Book2-7.1, 7.4.1,7.5.1,7.5.2)			
<b>Module-4 Test Planning, Management, Execution and Reporting (8 hours)</b>			
Test Planning: Preparing a test plan, Scope Management, Deciding the test approach, Setting up criteria for testing, Identifying responsibilities, Staffing, and Training Needs, Identifying Resource Requirements, Identifying Test Deliverables, Testing task, Test Management: Choice of standards, Test infrastructure management, Test people management, Test Process, Test Reporting. (Text Book1-15.2.1-15.2.8,15.3.1-15.3.3,15.4,15.5)			
<b>Module-5 Testing Tools, Metrics and Measurements (8 hours)</b>			
Automation and Testing tools: Need for automation, Categorization of Testing tools, Selection of Testing tools, Guidelines for Automated Testing, Test Metrics and Measurement: What are Metrics and Measurements?, Why metrics in Testing?, Types of metrics, Project metrics, Progress metrics, Productivity metrics (Text Book1-16.5,17.1, 17.2, 17.3, 17.4, 17.5, 17.6) (Text Book2-15.1,15.2,15.3,15.5)			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE452.1</b>	Derive test cases for any given problem.

<b>22CSE452.2</b>	Compare the different testing techniques.
<b>22CSE452.3</b>	Classify the problem into a suitable testing model.
<b>22CSE452.4</b>	Apply the appropriate technique for the design of flow graphs.
<b>22CSE452.5</b>	To prepare the Test plan and to produce the report.
<b>22CSE452.6</b>	Apply the appropriate Automated Testing Tools and Metrics.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Software testing: Principles and Practices	Gopalaswamy Ramesh, Srinivasan Desikan	Pearson	1 <sup>st</sup> Edition, 2007
2	Software testing: Principles and Practices	Naresh Chauhan	Oxford University Press	2 <sup>nd</sup> Edition, 2016
<b>Reference Books</b>				
1	Foundations of Software Testing	Aditya P Mathur	Pearson Education	2 <sup>nd</sup> Edition, 2013
2	Software Testing, A Craftsman's Approach	Paul C. Jorgensen	Auerbach Publications	4 <sup>th</sup> Edition, 2013
3	Software Testing and Analysis – Process, Principles and Techniques	Mauro Pezze, Michal Young	Wiley India	1 <sup>st</sup> Edition, 2008

**Web links and Video Lectures (e-Resources):**

- <https://www.softwaretestinghelp.com/selenium-tutorial-1/>
- <http://softwaretestingfundamentals.com/software-testing-methods/>
- [https://www.tutorialspoint.com/software\\_testing/software\\_testing\\_tutorial.pdf](https://www.tutorialspoint.com/software_testing/software_testing_tutorial.pdf)
- <https://nptel.ac.in/courses/106105150/>

**Course Articulation Matrix**

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
<b>22CSE452.1</b>			2											
<b>22CSE452.2</b>	2													
<b>22CSE452.3</b>			2		2									
<b>22CSE452.4</b>					2									
<b>22CSE452.5</b>					3					3				
<b>22CSE452.6</b>			2											

1: Low 2: Medium 3: High

<b>Python Programming Laboratory</b>			
Course Code	<b>22CSE46L</b>	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	20	Credits	01
<p><b>Course Learning Objectives:</b> The objective of the course is to</p> <ul style="list-style-type: none"> <li>• Learn the syntax and semantics of Python programming language.</li> <li>• Illustrate the process of structuring the data using lists, tuples and dictionaries.</li> <li>• Demonstrate the use of built-in functions to read/write files.</li> <li>• Interpret the concepts of Object-Oriented Programming as used in Python.</li> <li>• Understand the working of numpy and pandas.</li> <li>• Understanding the API creation and working with Postman</li> </ul>			
<b>PART A (Laboratory Programs)</b>			
<p>1 a) Write a program for comparing two numbers.  b) program to find LCM of two number using GCD  c )Write a program to display all the prime numbers within an interval</p>			
<p>2 a) Write a python program to check if the number is an Armstrong number or not.  b) Write a Program to display the Fibonacci sequence up to n-th term using recursion  c) Write programs for searching, splitting, and replacing strings based on pattern matching using regular expressions.</p>			
<p>3 a) <b>Collatz Sequence:</b> 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.  b) Python Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers</p>			
<p>4 a) Write a python program to accept N numbers from the user. Find and display sum of all even numbers and product of all odd numbers in entered list.  b) For a given list num=[45,22,14,65,97,72], write a python program to replace all the integers divisible by 3 with “ppp” and all integers divisible by 5 with “qqq” and replace all the integers divisible by both 3 and 5 with “pppqqq” and display the output.  c) Write a program to convert the binary number to decimal using function</p>			
<p>5 a) Write a 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}}</p> <p>Output:</p> <hr/> <pre> Number of things being brought: - Apples 7 - Cups 3 - Cakes 0 - Ham Sandwiches 3 - Apple Pies 1 </pre> <hr/> <p>b) Write a function named displayInventory(inventory) that would take any possible “inventory” (for example, { 'arrow': 12, 'gold coin': 42, 'rope': 1, 'torch': 6, 'dagger': 1 }) and display it like the following:</p>			



Inventory:

12 arrow

42 gold coin

1 rope

6 torch

1 dagger

Total number of items: 62

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:

45 gold coin

1 rope

1 ruby

1 dagger

Total number of items: 48

6 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 to create a class `Time`, display the time in `hh:mm:ss` format and add two time object by using the `__init__`, `__str__` and `__add__` methods.

7. Write a Python program to Create a User-Defined Exception by creating a `PercentageError` to derive it from the Exception base class. Derive three classes `InvalidPercentageError`, `LessPercentageError` and `checkPercentage` from `PercentageError` class. To decide which student can enroll in a particular course, there is a cut-off and maximum score. If a student's percentage is in the acceptable range, they are enrolled, or else an exception is raised. In this case, no student enrolling should have a percentage less than the cut-off, and their score cannot be higher than the maximum score. Use exception handling to identify the eligible student by assuming necessary class members.

8. Write a Python program to demonstrate the concept of polymorphism by creating a class `Shape` with a method `area()`. Create three classes, `Circle`, `Rectangle`, and `Triangle` by inheriting them from the `Shape` class with respective fields and constructors override the method `area()` in each derived class to calculate the area of respective shapes. Create instances of all three derived classes and display the area.

9. Write programs to create numpy arrays of different shapes and from different sources, reshape and slice arrays, add array indexes, and apply arithmetic, logic, and aggregation functions to some or all array elements

10. Write programs to use the pandas data structures: Frames and series as storage containers and for a variety of data-wrangling operations, such as:

- a. Single-level and hierarchical indexing
- b. Handling missing data
- c. Arithmetic and Boolean operations on entire columns and tables
- d. Database-type operations (such as merging and aggregation)
- e. Plotting individual columns and whole tables
- f. Reading data from files and writing data to files

11 Using the Postman perform the following operations
a. Create the first recipe using Postman.
b. Create the second recipe using Postman.
c. Retrieve all the recipes using Postman.
d. Set the recipes to published using Postman.
e. Retrieve all the recipes using Postman again.
f. Modify the recipe using Postman.
g. Get a specific recipe back using Postman.

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22CSE46L.1</b>	Demonstrate proficiency in handling of conditions, loops and creation of functions.
<b>22CSE46L.2</b>	Identify the methods to create and manipulate lists, tuples and dictionaries.
<b>22CSE46L.3</b>	Discover the commonly used operations involving file systems.
<b>22CSE46L.4</b>	Interpret the concepts of Object-Oriented Programming as used in Python.
<b>22CSE46L.5</b>	Identify the methods to work with numpy and pandas.
<b>22CSE46L.6</b>	Interpret the concept of API and working with Postman

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Automate the Boring Stuff with Python	Al Sweigart	No Starch Press	1 <sup>st</sup> Edition, 2015
2	Python API Development Fundamentals	Jack Chan, Ray Chung, Jack Huang	Packt Publishing	November 2019

<b>Web links and Video Lectures (e-Resources):</b>
<ul style="list-style-type: none"> <li>• <a href="https://www.youtube.com/watch?v=1F_OgqRuSdI&amp;list=PL0-84-yl1fUnRuXGFe_F7qSH1LEnn9LkWs">https://www.youtube.com/watch?v=1F_OgqRuSdI&amp;list=PL0-84-yl1fUnRuXGFe_F7qSH1LEnn9LkWs</a></li> </ul>

### Course Articulation Matrix

Course Outcomes (COs)	Program Outcomes (POs)													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
<b>22CSE46L.1</b>						2					2		2	
<b>22CSE46L.2</b>						2					2		2	
<b>22CSE46L.3</b>						2					2		2	
<b>22CSE46L.4</b>						2					2		2	
<b>22CSE46L.5</b>						2				2			2	
<b>22CSE46L.6</b>						2				2			2	

1: Low 2: Medium 3: High

<b>Universal Human Values- II</b>			
Course Code	<b>22UHV47</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	<b>02</b>	Exam Hours	02
<b>Course Learning Objectives:</b>			
<p>This introductory course input is intended:</p> <ol style="list-style-type: none"> <li>1. To help the students appreciate the essential complementarity between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.</li> <li>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.</li> <li>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.</li> </ol>			
<b>Module-1 Introduction to Value Education</b>			
<p>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.</p> <p>Activities: Sharing about Oneself, Exploring Human Consciousness and Exploring Natural Acceptance. <span style="float: right;"><b>5 Hours</b></span></p>			
<b>Module-2 – Harmony in the Human Being</b>			
<p>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.</p> <p>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. <span style="float: right;"><b>5 hours</b></span></p>			
<b>Module 3 – Harmony in the Family and Society</b>			
<p>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.</p> <p>Activities: Exploring the Feeling of Trust, Exploring the Feeling of Respect and Exploring the Feeling systems to fulfil Human Goal. <span style="float: right;"><b>5 hours</b></span></p>			
<b>Module-4 – Harmony in the Nature/Existence</b>			
<p>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.</p> <p>Activities: Exploring the Four Orders of Nature and Co-existence in Existence. <span style="float: right;"><b>5 hours</b></span></p>			
<b>Module-5 – Implications of the Holistic Understanding – a Look at Professional Ethics</b>			
<p>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</p> <p>Activities: Exploring Ethical Human Conduct, Humanistic Models in Education and steps of Transition towards Universal Human Order. <span style="float: right;"><b>5 hours</b></span></p>			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22UHV47.1</b>	Practice the method of self-exploration to understand the basic human aspiration.
<b>22UHV47.2</b>	Distinguish between needs of self and body.
<b>22UHV47.3</b>	Evolve a program for self-regulation and health.
<b>22UHV47.4</b>	Differentiate between the characteristics and activities of different orders and study the mutual fulfillment among them.
<b>22UHV47.5</b>	Realize sustainable solutions to the problems in society and nature.
<b>22UHV47.6</b>	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
<b>Textbooks</b>				
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
<b>Reference Books</b>				
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](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](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/1nh9m5ibEtvMyqekeiexAJtfbdNmtt6-?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=PLWDeKF97v9SP7wSlapZcQRrT7OH0ZIGC4>

### 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

<b>Biology for Engineers</b>			
Course Code	<b>22BFE47</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:0)	SEE Marks	50
Credits	<b>02</b>	Exam Hours	02
<b>Course Learning Objectives:</b>			
<ol style="list-style-type: none"> <li>1. To bring awareness of biological concepts to engineering students</li> <li>2. To introduce the building blocks of life and their complexity</li> <li>3. To encourage interdisciplinary studies and projects</li> <li>4. To appreciate the discoveries that mimic nature and its working</li> <li>5. To inculcate nature-inspired design and operational principles</li> </ol>			
<b>Module-1</b>			<b>5 Hours</b>
<b>Basic Cell Biology:</b> 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.			
<b>Module-2</b>			<b>5 Hours</b>
<b>Biochemistry and Molecular Aspects of Life:</b> 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.			
<b>Module-3</b>			<b>5 Hours</b>
<b>Bioinspired Engineering based on human physiology:</b> 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).			
<b>Module-4</b>			<b>5 Hours</b>
<b>Relevance of Biology as an interdisciplinary approach:</b> 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).			
<b>Module-5</b>			<b>5 Hours</b>
<b>Bioinspired Algorithms and Applications:</b> 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.			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22BFE47.1</b>	Discuss how the cell forms the basic building block of life
<b>22BFE47.2</b>	Distinguish between transcription and translation
<b>22BFE47.3</b>	Describe the role played by proteins within the cell
<b>22BFE47.4</b>	Analyze the role of bioinspired design in novel applications
<b>22BFE47.5</b>	Apply bioinspired design principles to other domains
<b>22BFE47.6</b>	Implement a simple genetic algorithm

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Text Books</b>				
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	5 <sup>th</sup> Edition, 2012
<b>Reference books</b>				
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	1 <sup>st</sup> Edition, 2016
4	A Practical Guide to Bio-inspired Design	Hashemi Farzaneh, Helena, Lindemann, Udo	Springer	2019

**Web links/Video Lectures/MOOCs**

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

<b>COMPUTATIONAL TOOLS FOR ENGINEERS</b>			
Course Code:	<b>22CTE48</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	<b>01</b>	Exam Hours	02
<b>Course Learning Objectives:</b>			
<ol style="list-style-type: none"> <li>1. Apply modeling and simulation tools for a wide range of engineering problems.</li> <li>2. Understand the analysis of data in Excel with statistics.</li> <li>3. Use MATLAB and Simulink to perform engineering system analysis.</li> </ol>			
<p>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.</p>			
<b>Module 1</b>			
<b>Engineering Design Analysis</b>			
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).			
			<b>10 Hours</b>
<b>Module 2</b>			
<b>Data Analysis with EXCEL</b>			
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.			
			<b>4 Hours</b>
<b>Module 3</b>			
<b>MATLAB and Simulink for Engineers</b>			
Applications of MATLAB and Simulink in electrical engineering, electrical machines and power system projects, simulation of rectifiers, inverters, choppers, and cycloconverters.			
			<b>10 Hours</b>
<b>Course Project</b>			
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).			

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



Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Reference Books</b>				
1	<b>MATLAB and Simulink for Engineers</b>	Agam Kumar Tyagi	Oxford University Press	2012
2	<b>Practical Finite Element Analysis</b>	Nitin S.Gokhale	Finite to Infinite	2020
3	<b>Excel Crash Course for Engineers</b>	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
<b>22CTE48.1</b>	1				1	1								
<b>22CTE48.2</b>		1			2				2					
<b>22CTE48.3</b>		1			2									
<b>22CTE48.4</b>					2	2								
<b>22CTE48.5</b>	1								2					

1: Low 2: Medium 3: High

<b>Industry Oriented Training - Business Etiquettes</b>			
Course Code	<b>22ITB49A</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02
<b>Course Learning Objectives:</b>			
11. Know the components of self-introduction 12. Develop a resume with the inclusion of core competencies 13. Involve and contribute to group discussions 14. Develop effective communication to succeed in the professional career 15. Know the etiquettes of digital communication			
<b>Module-1</b>			
<b>Self-Introduction &amp; Essentials of grooming</b>			
<b>Self-Introduction:</b> Learn the secret to introducing Yourself, Things to avoid when introducing yourself. Activity: Video record the self-introduction. <b>Essentials of grooming:</b> Creating the first impression, what does the well-dressed man wear? What does the well-dressed woman wear? Personal hygiene and habits. <b>4 Hours</b>			
<b>Module-2</b>			
<b>Resume Writing</b>			
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. <b>4 Hours</b>			
<b>Module-3</b>			
<b>Group Discussion</b>			
Types, process, Evaluation criteria, Do's and Don'ts Activity: Group discussions have to be held during the training sessions. <b>4 Hours</b>			
<b>Module-4</b>			
<b>Communicate effectively</b>			
Build a Story, Just a Minute, Group Activities, Team building activities, Role Play, Presentation Skills. <b>4 Hours</b>			
<b>Module-5</b>			
<b>Digital right and wrong</b>			
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. <b>4 Hours</b>			
<b>Course Outcomes:</b> At the end of the course the student will be able to:			
<b>22ITB49A.1</b>	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		
<b>22ITB49A.2</b>	Develop a resume inclusive of core competencies, and action verbs which are compatible with Applicant Tracking Systems		
<b>22ITB49A.3</b>	Demonstrate the types, process and evaluation process of Group Discussion and carry out effective group discussions		
<b>22ITB49A.4</b>	Develop skills required for effective communication		
<b>22ITB49A.5</b>	Associate and be accustomed to the etiquette to be followed during online meetings		

<b>Sources</b>	
1.	English for Common Interactions in the Workplace: Basic Level: Coursera: <a href="https://www.coursera.org/learn/english-common-interactions-workplace-basic-level">https://www.coursera.org/learn/english-common-interactions-workplace-basic-level</a>
2.	Personal Communication-Introduce Yourself With Confidence: <a href="https://www.udemy.com/course/how-to-introduce-yourself/">https://www.udemy.com/course/how-to-introduce-yourself/</a>
3.	Professionalism, Grooming and Etiquette: <a href="https://www.edx.org/course/professionalism-grooming-and-etiquette">https://www.edx.org/course/professionalism-grooming-and-etiquette</a>
4.	How to Write a Resume: <a href="https://www.coursera.org/learn/how-to-write-a-resume#syllabus">https://www.coursera.org/learn/how-to-write-a-resume#syllabus</a>
5.	Group Discussion Strategies: <a href="https://www.udemy.com/course/group-discussion-strategies/">https://www.udemy.com/course/group-discussion-strategies/</a>
6.	Communication Strategies for a Virtual Age: <a href="https://www.coursera.org/learn/communication-strategies-virtual-age#syllabus">https://www.coursera.org/learn/communication-strategies-virtual-age#syllabus</a>
<b>References</b>	
1.	<a href="https://simplifytraining.com/course/personal-hygiene-and-good-grooming/">https://simplifytraining.com/course/personal-hygiene-and-good-grooming/</a>
2.	<a href="https://www.udemy.com/course/group-discussion-strategies/">https://www.udemy.com/course/group-discussion-strategies/</a>
3.	<a href="https://www.educba.com/course/group-discussion/">https://www.educba.com/course/group-discussion/</a>
4.	<a href="https://getrafiki.ai/meetings/rules-of-virtual-meeting-etiquette-every-sales-professional-should-follow/">https://getrafiki.ai/meetings/rules-of-virtual-meeting-etiquette-every-sales-professional-should-follow/</a>
5.	<a href="https://thedigitalworkplace.com/articles/online-meeting-etiquette-for-attendees/">https://thedigitalworkplace.com/articles/online-meeting-etiquette-for-attendees/</a>
6.	<a href="https://rigorousthemes.com/blog/virtual-meeting-etiquette-guidelines-ground-rules/">https://rigorousthemes.com/blog/virtual-meeting-etiquette-guidelines-ground-rules/</a>

### 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

<b>Industry Oriented Training - Computing Skills</b>			
Course Code	<b>22ITC49B</b>	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	-	Exam Hours	02
<b>Course Learning Objectives:</b>			
6. Use logical conditions for problem-solving and also introduce the concepts of arrays 7. Know functions, function calls, and parameter passing 8. Introduce algorithms and appreciate their importance in problem-solving 9. Introduce the core concepts of OOP's 10. Differentiate between front-end & back-end development and recognize the use of database management			
<b>Module-1</b>			
<b>Introduction to computing constructs</b>			
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.			
<b>4 Hours</b>			
<b>Module-2</b>			
<b>Functions &amp; Pointers</b>			
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.			
<b>4 Hours</b>			
<b>Module-3</b>			
<b>Algorithm analysis</b>			
Introduction to Algorithm Analysis, Big-O, Big-O Examples, Dynamic Array Operations, Bubble Sort, Selection Sort, Insertion Sort, Recursion, Recursive Binary Search, Merge Sort.			
<b>4 Hours</b>			
<b>Module-4</b>			
<b>Object-oriented programming</b>			
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.			
<b>4 Hours</b>			
<b>Module-5</b>			
<b>Frontend and backend development</b>			
UI, Database management: DBMS overview, Relational Data Model and the CREATE TABLE Statement, Basic Query Formulation with SQL.			
<b>4 Hours</b>			

<b>Course Outcomes:</b> At the end of the course the student will be able to:	
<b>22ITC49B.1</b>	Illustrate the use of logical conditions, declare and manipulate data into arrays
<b>22ITC49B.2</b>	Implement functions, function calls, and parameter passing
<b>22ITC49B.3</b>	Design, implement, and evaluate an algorithm to meet desired needs
<b>22ITC49B.4</b>	Describe the core concepts of OOP's
<b>22ITC49B.5</b>	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
<b>22ITC49B.1</b>	2	1	1											
<b>22ITC49B.2</b>	2	1	1											
<b>22ITC49B.3</b>	1	1	2											
<b>22ITC49B.4</b>	2		1											
<b>22ITC49B.5</b>	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](mailto:sjec@sjec.ac.in) | Website: [www.sjec.ac.in](http://www.sjec.ac.in)

