B.E. in Computer Science and Engineering Scheme of Teaching and Examinations Admitted Batch 2023-2024

III SEMESTER

Sl. No	Course	Course Code	Course Title	Teaching Department (TD) and Question Paper Setting Board (PSB)	Teaching Hours/Week				Examination				Credits
					Theory Lecture	Tutorial	Practical/ Drawing	SDA	Duration in hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PCC/ BSC	22UMA311C	Mathematics for Computer Science	TD: Maths PSB: Math CS	3	2	0		03	50	50	100	4
2	IPCC	22UCS312C	Digital Design & Computer Organization	TD:CS PSB:CS	3	0	2		03	50	50	100	4
3	IPCC	22UCS313C	Operating Systems	TD:CS PSB:CS	3	0	2		03	50	50	100	4
4	PCC	22UCS314C	Data Structures and Applications	TD:CS PSB:CS	3	0	0		03	50	50	100	3
5	PCCL	22UCS317L	Data Structures Lab	TD:CS PSB : CS	0	0	2		03	50	50	100	1
6	ESC	22UCS316C 22UCS326C	ESC/ETC/PLC	TD:CS PSB:CS	2	0	2		03	50	50	100	3
7	UHV	22UHS317L	Social Connect and Responsibility	Any Department	0	0	2		01	100	---	100	1
8	AEC/ SEC	22UCS318L 22UCS328L 22UCS338L 22UCS348L	Ability Enhancement Course/Skill Enhancement Course - III	TD: Concerned department PSB:CS	If the course is a Theory 1	0	0		01	50	50	100	1
					If a course is a laboratory 0	0	2		02				
9	MC	22UHS002M	National Service Scheme(NSS)	NSS coordinator	0	0	2			100	---	100	0
		BPEK359	Physical Education(PE) (Sports and Athletics)	Physical Education Director									
		BYOK359	Yoga	Yoga Teacher									
									Total	**550**	**350**	**900**	**21**

Course Code: 22UMA311C	**Mathematics for Computer Science**	**Semester:3**
		Credits: 04
L:T:P - 4:0:0:0		CIE Marks: 50
Total Hours/Week: 50		SEE Marks: 50

Course objectives: This course will enable the students:

1. To find the association between attributes and the correlation between two variables
2. To introduce the concept of random variables, probability distributions, specific discrete and continuous distributions with practical application in Computer Science Engineering and social life situations.
3. To Provide the principles of statistical inferences and the basics of hypothesis testing with Emphasis on some commonly encountered hypotheses.

Module-1: Curve fitting, Correlation and Regressions — 10 Hrs.

Principles of least squares, Curve fitting by the method of least squares in the form $y = a + bx$, $y = a + bx + cx^2$, and $y = ab^x$. Correlation, Co-efficient of correlation, Lines of regression, Angle between regression lines, rank correlation.

Module-2: Probability Distributions — 10 Hrs.

Probability Distributions: Review of basic probability theory. Random variables (discrete and continuous), probability mass and density functions. Mathematical expectation, mean and variance. Binomial, Poisson and normal distributions- problems (derivations for mean and standard deviation for Binomial and Poisson distributions only)-Illustrative examples

Module-3: Joint probability distribution & Markov Chain — 10 Hrs.

Joint probability distribution: Joint Probability distribution for two discrete random variables, expectation, covariance and correlation.

Markov Chain: Introduction to Stochastic Process, Probability Vectors, Stochastic matrices, Regular stochastic matrices, Markov chains, Higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states.

Module-4: Statistical Inference 1 — 10 Hrs.

Introduction, sampling distribution, standard error, testing of hypothesis, levels of significance, test of significances, confidence limits, simple sampling of attributes, test of significance for large samples, comparison of large samples.

Module-5: Statistical Inference 2 — 10 Hrs.

Sampling variables, central limit theorem and confidences limit for unknown mean. Test of Significance for means of two small samples, students ‘t’ distribution, Chi-square distribution as a test of goodness of fit. F-Distribution.

Reference Books

1. **Ronald E.Walpole, RaymondHMyers, SharonLMyers&KeyingYe** “Probability & Statistics for Engineers & Scientists”,Pearson Education, 9thedition, 2017.
2. **PeterBruce,Andrew Bruce & Peter Gedeck**“Practical Statistics for Data Scientists”O’ ReillyMedia, Inc., 2nd edition **2020**.

Course Outcomes

After completion of the course student will be able to

1. Make use of correlation and regression analysis to fit a suitable mathematical model for statistical data.
2. Explain the basic concepts of probability and probability distribution.
3. Apply the notion of a discrete-time Markov chain and n-step transition probabilities to solve the given problem.
4. Compute the confidence intervals for the mean of the population.
5. Apply the Statistical Inference related to engineering problems.

Course Outcomes	**Programme Outcomes**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CO1	3	2	--	--	--	--	--	--	--	--	--	--
CO2	3	2	--	--	--	--	--	--	--	--	--	--
CO3	3	2	--	--	--	--	--	--	--	--	--	--
CO4	3	2	--	--	--	--	--	--	--	--	--	--
CO5	3	2	--	--	--	--	--	--	--	--	--	--

Course Code: **22UCS312C**	**DIGITAL DESIGN AND COMPUTER ORGANIZATION**	**Semester:3** Credits :04
Hours/Week: 03		CIE Marks: 50
Total Hours of Pedagogy : 40 Hours Theory + 20 Hours of Practical		SEE Marks:50
Course Type : Integrated		

Course Objectives:

- To demonstrate the functionalities of binary logic system.
- To explain the working of combinational and sequential logic systems.
- To realize the basic structure of computer system.
- To illustrate the working of I/O operations and processing unit.

Module-1	**8 Hrs.**
Introduction to Digital Design: Binary Logic, Basic Theorems And Properties Of Boolean Algebra, Boolean Functions, Digital Logic Gates, Introduction, The Map Method, Four-Variable Map, Don't-Care Conditions, NAND and NOR Implementation, Other Hardware Description Language – Verilog Model of a simple circuit. **Text book 1: 1.9, 2.4, 2.5, 2.8, 3.1, 3.2, 3.3, 3.5, 3.6, 3.9**	
Module-2	**8 Hrs.**
Combinational Logic: Introduction, Combinational Circuits, Design Procedure, Binary Adder- Subtractor, Decoders, Encoders, Multiplexers. HDL Models of Combinational Circuits – Adder, Multiplexer, Encoder. **Sequential Logic**: Introduction, Sequential Circuits, Storage Elements: Latches, Flip-Flops. **Text book 1: 4.1, 4.2, 4.4, 4.5, 4.9, 4.10, 4.11, 4.12, 5.1, 5.2, 5.3, 5.4.**	
Module-3	**8 Hrs.**
Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus structure, Performance Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. **Machine Instructions and Programs:** Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Addressing Modes. **Text book 2: 1.2, 1.3, 1.4, 1.6, 2.2, 2.3, 2.4, 2.5**	
Module-4	**8 Hrs.**
Input/output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Direct Memory Access: Bus Arbitration, Speed, size and Cost of memory systems. Cache Memories – Mapping Functions. **Text book 2: 4.1, 4.2.1, 4.2.2, 4.2.3, 4.4, 5.4, 5.5.1**	
Module-5	**8 Hrs.**
Basic Processing Unit: Some Fundamental Concepts: Register Transfers, Performing ALU operations, Fetching a word from Memory, Storing a word in memory. Execution of a Complete Instruction. **Pipelining:** Basic concepts, Role of Cache memory, Pipeline Performance. **Text book 2: 7.1, 7.2, 8.1**	
Practical Module	

Sl. No.	**Experiments**

	(Note: Experiments Simulation packages preferred: Multisim, Modelsim, PSpice or any other relevant)
1.	Given a 4-variable logic expression, simplify it using appropriate technique and simulate the same using basic gates.
2.	Design a 4 bit full adder and subtractor and simulate the same using basic gates.
3.	Design Verilog HDL to implement simple circuits using structural, Data flow and Behavioural model.
4.	Design Verilog HDL to implement Binary Adder-Subtractor – Half and Full Adder, Half and Full Subtractor.
5.	Design Verilog HDL to implement Decimal adder
6.	Design Verilog program to implement Different types of multiplexer like 2:1, 4:1 and 8:1.
7.	Design Verilog program to implement types of De-Multiplexer.
8.	Design Verilog program for implementing SR, JK and D Flip-Flops.

Suggested Learning resources

1. M. Morris Mano & Michael D. Ciletti, Digital Design With an Introduction to Verilog Design, 5E, Pearson Education.
2. Carl Hamacher, ZvonkoVranesic, SafwatZaky, Computer Organization, 5th Edition, Tata McGraw Hill.

Course Outcomes:

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

CO1: Apply the Boolean Laws and K–Map techniques to simplify Boolean expressions.

CO2: Design different types of combinational and sequential circuits.

CO3: Describe the fundamentals of machine instructions, addressing modes and Processor performance.

CO4: Explain the approaches involved in achieving communication between processor and I/O devices.

CO5: Analyze internal Organization of Memory and Impact of cache/Pipelining on Processor Performance.

Course Outcomes	**Programme Outcomes**												**Program Specific Outcomes**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	3	1	1		1								1		3
CO2	1	2	3	1	1							1	1		3
CO3	1	1	1										1		1
CO4	1	1	1										1		1
CO5	1	2	2										1		2

Course Code 22UCS313C	**OPERATING SYSTEMS**	**Semester:3**
		Credits :04
Hours/Week: (3_0_2_0)		**CIE Marks : 50**
Total Hours of pedagogy 40 hours Theory + 20 hours Practical		**SEE Marks : 100**
Course Type: Integrated		

Course Objectives:

- To Demonstrate the need for OS and different types of OS
- To discuss suitable techniques for management of different resources
- To demonstrate different APIs/Commands related to processor, memory, storage and file system management.

Module-1	**08 Hrs.**

Introduction to operating systems: System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments.

Operating System Services: User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System debugging, Operating System generation; System boot.

Textbook 1: Chapter – 1 (1.1-1.12), 2 (2.2-2.11)

Module-2	**08 Hrs.**

Process Management: Process concept; Process scheduling; Operations on processes; Inter process communication.

Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues.

Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Thread scheduling; Multiple-processor scheduling,

Textbook 1: Chapter – 3 (3.1-3.4), 4 (4.1-4.4), 5 (5.1 -5.5)

Module-3	**08Hrs.**

Process Synchronization: Synchronization: The critical section problem; Peterson’s solution; Synchronization hardware; Semaphores; Classical problems of synchronization.

Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.

Textbook 1: Chapter – 6 (6.1-6.6), 7 (7.1 -7.7)

Module-4	**08Hrs.**

Memory Management: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.

Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

Textbook 1: Chapter -8 (8.1-8.6), 9 (9.1-9.6)

Module-5	**08Hrs.**

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

Secondary Storage Structure, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; **Protection**: Goals of protection, Principles of protection, Domain of protection, Access matrix.

Textbook 1: Chapter – 10 (10.1-10.5) ,11 (11.1-11.5),12 (12.1-12.5), 14 (14.1-14.4)

Practical Module

Sl.NO	Experiments
1	Develop a c program to implement the Process system calls (fork (), exec(), wait(), create process,terminate process)
2	Simulate the following CPU scheduling algorithms to find turnaround time and waiting time a) FCFS. b) SJF c) Round Robin d) Priority.
3	Develop a C program to simulate producer-consumer problem using semaphores.
4	Develop a C program which demonstrates interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close APIs in your program.
5	Develop a C program to simulate Bankers Algorithm for Deadlock Avoidance.
6	Develop a C program to simulate the following contiguous memory allocation Techniques: a) Worst fit b) Best fit c) First fit.
7	Develop a C program to simulate page replacement algorithms: a) FIFO b) LRU
8	Simulate following File Organization Techniques a) Single level directory b) Two level directory
9	Develop a C program to simulate the Linked file allocation strategies.
10	Develop a C program to simulate SCAN disk scheduling algorithm.

Suggested Learning resources

Textbooks

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 8th edition, Wiley-India, 2015

Reference Books

1. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.

William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

Course Outcomes:

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

CO1: Explain the structure and functionalities of operating system

CO2: Apply appropriate CPU scheduling algorithms for the given problem, memory management techniques

CO3: Analyse the various techniques for process synchronization and deadlock handling, memory management., File system management.

CO4: Describe the need for information protection mechanisms

CO5: Simulate simple functionalities , policies of Operating system.

Course Outcomes	Programme Outcomes												Program Specific Outcomes		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	1	1	1									1	1		2
CO2	1	2	3	1								1	1		2
CO3	2	3	3	1								1	1		2
CO4	1	1	1	1								1	1		2
CO5	1	2	3	1								1	1		2

Sub. Code: 22UCS314C	**DATA STRUCTURES AND APPLICATIONS**	**Semester : 3**
L: T: P :: 3: 0: 0		**Credits : 3**
No. of Lecture Hours: 40		**CIE Marks :50**
Course Type: Theory		**SEE Marks :50**

Course Objectives:

1. To explain fundamentals of data structures and their applications.
2. To illustrate representation of Different data structures such as Stack, Queues, Linked Lists, Trees and Graphs.
3. To Design and Develop Solutions to problems using Linear Data Structures
4. To discuss applications of Nonlinear Data Structures in problem solving.
5. To introduce advanced Data structure concepts such as Hashing and Optimal Binary Search Trees

Module-1	**08 Hrs.**

INTRODUCTION TO DATA STRUCTURES: Data Structures, Classifications (Primitive & Non-Primitive), Data structure Operations Review of pointers and dynamic Memory Allocation, ARRAYS and STRUCTURES: Arrays, Dynamic Allocated Arrays, Structures and Unions, Polynomials, Sparse Matrices, representation of Multidimensional Arrays, Strings

Text Book: Chapter-1:1.2 , Chapter-2: 2.1 to 2.7
Reference Book 1: 1.1 to 1.4

Module-2	**08 Hrs.**

STACKS: Stacks, Stacks Using Dynamic Arrays, Evaluation and conversion of Expressions, QUEUES: Queues, Circular Queues, Using Dynamic Arrays, Multiple Stacks and queues.

Text Book: Chapter-3: 3.1,3.2, 3.6, 3.3, 3.4, 3.7

Module-3	**08 Hrs.**

LINKED LISTS: Singly Linked, Lists and Chains, Representing Chains in C, Linked Stacks and Queues, Polynomials, Additional List Operations, Sparse Matrices, Doubly Linked List.

Text Book: Chapter-4: 4.1 to 4.4, 4.5,4.7,4.8

Module-4	**08 Hrs.**

TREES: Introduction, Binary Trees, Binary Tree Traversals, Threaded Binary Trees. Binary Search trees, Selection Trees, Forests, Representation of Disjoint sets, Counting Binary Trees.

Text Book: Chapter-5: 5.1 to 5.3, 5.5, 5.7 to 5.11

Module-5	**08 Hrs.**

PRIORITY QUEUES: Single and double ended Priority Queues, Leftist Trees INTRODUCTION TO EFFICIENT BINARY SEARCH TREES: Optimal Binary Search Trees, GRAPHS: The Graph Abstract Data Types, Elementary Graph Operations.

Text Book: Chapter 9: 9.1, 9.2 Chapter 10: 10.1, Chapter-6: 6.1, 6.2

Suggested Learning resources

- http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS35.html
- https://nptel.ac.in/courses/106/105/106105171/
- http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html
- https://www.youtube.com/watch?v=3Xo6P_V-qns&t=201s
- https://ds2-iiith.vlabs.ac.in/exp/selection-sort/index.html
- https://nptel.ac.in/courses/106/102/106102064/

- https://ds1-iiith.vlabs.ac.in/exp/stacks-queues/index.html
- https://ds1-iiith.vlabs.ac.in/exp/linked-list/basics/overview.html
- https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html
- https://ds1-iiith.vlabs.ac.in/exp/tree-traversal/index.html
- https://ds1-iiith.vlabs.ac.in/exp/tree-traversal/depth-first-traversal/dft-practice.html
- https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01350159542807756812559/overview

Textbook:

1. Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014

Reference Books:

1. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1 st Ed, McGraw Hill, 2014.
2. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2 nd Ed, Cengage Learning, 2014.
3. Reema Thareja, Data Structures using C, 3 rd Ed, Oxford press, 2012.
4. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Ed, McGraw Hill, 2013
5. A M Tenenbaum, Data Structures using C, PHI, 1989 6. Robert Kruse, Data Structures and Program Design in C, 2 nd Ed, PHI, 1994

Course Outcomes:

CO1: Explain the fundamental concepts of arrays, pointers, structures, unions, dynamic memory allocation, and hashing, and demonstrate their usage in programming contexts.

CO2: Illustrate the organization and working principles of linear data structures (such as stacks and queues) and nonlinear data structures (such as trees)

CO3: Design and implement programs using linear and nonlinear data structures to solve computational problems effectively.

CO4: Analyze problem requirements to identify and justify the choice of appropriate data structures for optimized solutions.

CO5: Develop efficient and scalable solutions by applying data structure techniques to model and solve real-world problems.

CO, PO and PSO Mapping Table:

Course Outcomes	Programme Outcomes												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	2	1	2								1	3		2
CO2	3	3	2	2								1	2		2
CO3	3	3	3	3									3		3
CO4	3	2	2	3								2	3		3
CO5	3	3	3	3								3	3		3

Course Code: 22UCS317L	**DATA STRUCTURES LABORATORY**	Semester	03
Credit: 01		CIE Marks	50
Number of Contact Hours/Week: 0:0:2		SEE Marks	50
Total Number of Lab Contact Hours: 28		Exam Hours	03

Course Learning Objectives:

This laboratory course enables students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Dynamic memory management
- Linear data structures and their applications such as stacks, queues and lists
- Non-Linear data structures and their applications such as trees and graphs

Descriptions (if any):

- Implement all the programs in “C” Programming Language and Linux OS.

Programs List:

1. Develop a Program in C for the following:
   a. Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), the second field is the date of the Day (A integer), the third field is the description of the activity for a particular day (A dynamically allocated String).
   b. Write functions create (), read () and display (); to create the calendar, to read the data from the keyboard and to print weeks activity details report on screen.

2. Develop a Program in C for the following operations on Strings.
   a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)
   b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR

   Support the program with functions for each of the above operations. Don't use Built-in functions.

3. Develop a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)
   a. Push an Element on to Stack
   b. Pop an Element from Stack
   c. Demonstrate how Stack can be used to check Palindrome
   d. Demonstrate Overflow and Underflow situations on Stack
   e. Display the status of Stack
   f. Exit

   Support the program with appropriate functions for each of the above operations

4. Develop 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: **+, -, *, /, %** (Remainder), **^** (Power) and alphanumeric operands.

5. Develop a Program in C for the following Stack Applications
   a. Evaluation of Suffix expression with single digit operands and operators: **+, -, *, /, %, ^**
   b. Solving Tower of Hanoi problem with n disks

6.	Develop a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations.
7.	Develop a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: ***USN, Name, Programme, Sem, PhNo*** a. Create a SLL of N Students Data by using *front insertion*. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion / Deletion at End of SLL d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack) e. Exit
8.	Develop a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: ***SSN, Name, Dept, Designation, Sal, PhNo*** a. Create a DLL of N Employees Data by using *end insertion*. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Exit
9.	Develop a Program in C for the following operationson Singly Circular Linked List (SCLL) with header nodes a. Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2y^2z - 4yz^5 + 3x^3yz + 2xy^5z - 2xyz^3$ b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations.
10.	Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit

Laboratory Outcomes: The student should be able to:

- CO1: Analyze various linear and non-linear data structures
- CO2: Demonstrate the working nature of different types of data structures and their applications
- Co3: Use appropriate searching and sorting algorithms for the give scenario.
- CO4: Apply the appropriate data structure for solving real world problems

CO, PO and PSO Mapping Table:

Course Outcomes	Programme Outcomes (POs)												Program Specific Outcomes (PSOs)		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1		3	3	3	3								3		3
CO2		3	3	3	3							3	3		3
CO3		3	3	3	3							2	3		3
Co4		3	3	3	3							2	3		3

Semester: 3	**OBJECT ORIENTED PROGRAMMING WITH C++**	**Course Code: 22UCS326C**
Credits: 3		LTP: 2-0-2
No. of teaching Hours: Lecture: 28 Theory+: 20 Practical		CIE Marks: 50
Course Type: Integrated		SEE Marks: 50

Course Objectives:

1: To understand object-oriented programming using C++and Gain knowledge about the capability to store information together in an object.

2: To illustrate the capability of a class to rely upon another class and functions.

3 : To Create and process data in files using file I/O functions

4. To understand the generic programming features of C++ including Exception handling

Module-1	**5 Hrs.**
An overview of C++: What is object-Oriented Programming? Introducing C++ Classes, The General Form of a C++ Program. Classes and Objects: Classes, Friend Functions, Friend lasses, Inline Functions, Parameterized Constructors, Static Class Members, When Constructors and Destructors are Executed, The Scope Resolution Operator, Passing Objects to functions, Returning Objects, Object Assignment Ch 11, Ch 12	
Module-2	**6 Hrs.**
Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects, Pointers to Objects, The this Pointer, Pointers to derived types, Pointers to class members. Functions Overloading, Copy Constructors: Functions Overloading, Overloading Constructor Functions. Copy Constructors, Default Function Arguments, Function Overloading and Ambiguity. Ch 13, Ch 14	
Module-3	**6 Hrs.**
Operator Overloading: Creating a Member Operator Function, Operator Overloading Using a Friend Function, Overloading new and delete Inheritance: Base-Class Access Control, nheritance and Protected Members, Inheriting Multiple Base Classes , Constructors, estructors and Inheritance, Granting Access, Virtual Base Classes Ch 15, Ch 16	
Module-4	**5 Hrs.**
Virtual Functions and Polymorphism: Virtual Functions, The Virtual Attribute is Inherited, Virtual Functions are Hierarchical, Pure Virtual Functions, Using Virtual Functions, Early vs Late Binding. Templates: Generic Functions, Applying Generic Functions, Generic Classes. The type name and export Keywords. The Power of Templates Ch 17, Ch 18	
Module-5	**6Hrs.**
Exception Handling: Exception Handling Fundamentals, Handling Derived-Class Exceptions, Exception Handling Options, Applying Exception Handling. The C++ I/O System Basics: C++ Streams, The C++ Classes, Formatted I/O File I/O: &lt;fstream&gt; and File Classes, Opening and Closing a File, Reading and Writing Text Files, Detecting EOF. Ch 19, Ch 20, Ch 21	
Practical Module	
1. Develop a C++ program to find the largest of three numbers	

2. Develop a C++ program to sort the elements in ascending and descending order.Develop an ALP to find the sum of first 10 integer numbers.
3. Develop a C++ program using classes to display student name, roll number, marks obtained in two subjects and total score of student.
4. Develop a C++ program for a bank empolyee to print name of the employee, account_no. &amp; balance.
5. Develop a C++ program to demonstrate function overloading for the following prototypes.
   add(int a, int b)
   add(double a, double b.
6. Develop a C++ program using Operator Overloading for overloading Unary minus operator..
7. Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers
8. Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values..
9. Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.
10. Develop a C++ program to write and read time in/from binary file using fstream
11. Develop a function which throws a division by zero exception and catch it in catch block. Write a C++ program to demonstrate usage of try, catch and throw to handle exception.
12. Develop a C++ program that handles array out of bounds exception using C++.

Reference Books:

1. Herbert schildt, The Complete Reference C++, 4 th edition, TMH, 2005
2. Balagurusamy E, Object Oriented Programming with C++, Tata McGraw Hill Education Pvt.Ltd., Sixth Edition 2016.
3. Bhave, “Object Oriented Programming With C++”, Pearson Education, 2004.

Course Outcomes:

1. Illustrate the basic concepts of object-oriented programming.
2. Design appropriate classes for the given real-world scenario.
3. Apply the knowledge of compile-time / run-time polymorphism to solve the given problem
4. Use the knowledge of inheritance for developing optimized solutions
5. Apply the concepts of templates and exception handling for the given problem
6. Use the concepts of input output streams for file operations

Course Outcomes	**Programme Outcomes**												**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
CO1	2		3										1		
CO2	1		3	3									2		
CO3	2	2											3		
CO4	2	2		3	1									2	
CO5	1		3												
CO6	1	2											1		

COURSE CODE: 22UHS317L	**SOCIAL CONNECT AND RESPONSIBILITY**	Credits: 01
		Semester:3
L:T:P - 0:0:3:1		CIE Marks: 50
Total Hours/Week: 15		SEE Marks: 50

Module-I	**03 Hrs.**

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

Module-II	**03 Hrs.**

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

Module-III	**03 Hrs.**

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

Module-IV	**03 Hrs.**

Water conservation: Knowing the present practices in the surrounding villages and implementation in the campus, documentary or photoblog presenting the current practices – Objectives, Visit, case study, report, outcomes.

Module-V	**03 Hrs.**

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

Reference Books:

1. Author/s last Name, initial (Year), Book Title (edition), Publisher

Course Outcomes:

After completion of the course student will be able to

1: Communicate and connect to the surrounding.
2: Create a responsible connection with the society
3: Involve in the community in general in which they work.
4: Notice the needs and problems of the community and involve them in problem –solving.
5: Develop among themselves a sense of social & civic responsibility & utilize their knowledge in finding practical solutions to individual and community problems.

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

CO5															

Course Code 22UCS328L	**Data analytics with Excel**	Sem:03
Hours/Week: 0:0:2:0		Credits :01
Total Hours of pedagogy:30 hrs		CIE Marks: 50
Course Type: Theory/Practical/ Integrated: Practical		SEE Marks: 50

Course Objectives:

- To Apply analysis techniques to datasets in Excel
- Learn how to use Pivot Tables and Pivot Charts to streamline your workflow in Excel
- Understand and Identify the principles of data analysis
- Become adept at using Excel functions and techniques for analysis
- Build presentation ready dashboards in Excel

Practical Module

Sl. No.	Experiment
1	Getting Started with Excel: Creation of spread sheets, Insertion of rows and columns, Drag & Fill, use of Aggregate functions.
2	Working with Data : Importing data, Data Entry & Manipulation, Sorting & Filtering.
3	Working with Data: Data Validation, Pivot Tables & Pivot Charts.
4	Data Analysis Process: Conditional Formatting, What-If Analysis, Data Tables, Charts & Graphs.
5	Cleaning Data with Text Functions: use of UPPER and LOWER, TRIM function, Concatenate.
6	Cleaning Data Containing Date and Time Values: use of DATEVALUE function, DATEADD and DATEDIF, TIMEVALUE functions.
7	Conditional Formatting: formatting, parsing, and highlighting data in spreadsheets during data analysis.
8	Working with Multiple Sheets: work with multiple sheets within a workbook is crucial for organizing and managing data, perform complex calculations and create comprehensive reports.
9	Create worksheet with following fields: Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP). Use appropriate formulas to calculate the above scenario. Analyse the data using appropriate chart and report the data.
10	Create worksheet on Inventory Management: Sheet should contain Product code, Product name, Product type, MRP, Cost after % of discount, Date of purchase. Use appropriate formulas to calculate the above scenario. Analyse the data using appropriate chart and report the data.
11	Create worksheet on Sales analysis of Merchandise Store: data consisting of Order ID, Customer ID, Gender, age, date of order, month, online platform, Category of product, size, quantity, amount, shipping city and other details. Use of formula to segregate different categories and perform a comparative study using pivot tables and different sort of charts.
12	Generation of report & presentation using Autofilter &macro.

Suggested Learning Resources:

- Berk & Carey - Data Analysis with Microsoft® Excel: Updated for Office 2007®, Third Edition, © 2010 Brooks/Cole, Cengage Learning, ISBN-13: 978-0-495-39178-4
- Wayne L. Winston - Microsoft Excel 2019: Data Analysis And Business Modeling, PHI, ISBN: 9789389347180
- Aryan Gupta - Data Analysis in Excel: The Best Guide. (https://www.simplilearn.com/tutorials/excel-tutorial/data-analysis-excel)

Course Outcomes:

CO1:	Use advanced functions and productivity tools to assist in developing worksheets.
CO2:	Manipulate data lists using Outline and PivotTables.
CO3:	Use Consolidation to summarize and report results from multiple worksheets.
CO4:	Apply Macros and Auto filter to solve the given real world scenario.

Course Outcomes	**Programme Outcomes**												**Program Specific Outcomes**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	3	3	3	3	3	1				2	1	2	2	1	2
CO2	3	3	3	3	3	1				2	1	2	2	1	2
CO3	3	3	3	3	3	1				2	1	2	2	1	2
CO4	3	3	3	3	3	1				2	1	2	2	1	2

SUBJECT CODE: 22UCS318L	**R Programming**	**Semester:3**
		Credits: 01
L:T:P - **0:0:2**		CIE Marks: 50
Total Hours/Week: **24**		SEE Marks: 50

Course objectives:

- To explore and understand how R and R Studio interactive environment.
- To understand the different data Structures, data types in R.
- To learn and practice programming techniques using R programming.
- To import data into R from various data sources and generate visualizations.
- To draw insights from datasets using data analytics techniques.

Practical Module

SL.NO	**Experiments**
1	Demonstrate the steps for installation of R and R Studio. Perform the following: a. Assign different type of values to variables and display the type of variable. b. Assign different types such as Double, Integer, Logical, Complex and Character and understand the difference between each data type. c. Demonstrate Arithmetic and Logical Operations with simple examples. d. Demonstrate generation of sequences and creation of vectors. e. Demonstrate Creation of Matrices f. Demonstrate the Creation of Matrices from Vectors using Binding Function. g. Demonstrate element extraction from vectors, matrices and arrays
2	Assess the Financial Statement of an Organization being supplied with 2 vectors of data: Monthly Revenue and Monthly Expenses for the Financial Year. You can create your own sample data vector for this experiment) Calculate the following financial metrics: a. Profit for each month. b. Profit after tax for each month (Tax Rate is 30%). c. Profit margin for each month equals to profit after tax divided by revenue. d. Good Months – where the profit after tax was greater than the mean for the year. e. Bad Months – where the profit after tax was less than the mean for the year. f. The best month – where the profit after tax was max for the year. g. The worst month – where the profit after tax was min for the year. **Note:** a. All Results need to be presented as vectors b. Results for Dollar values need to be calculated with $0.01 precision, but need to be presented in Units of $1000 (i.e 1k) with no decimal points c. Results for the profit margin ratio need to be presented in units of % with no decimal point. d. It is okay for tax to be negative for any given month (deferred tax asset) e. Generate CSV file for the data.
3	Develop a program to create two 3 X 3 matrices A and B and perform the following operations: a. Transpose of the matrix b. addition c. subtraction d. multiplication
4	Develop a program to find the factorial of given number using recursive function calls.
5	Develop an R Program using functions to find all the prime numbers up to a specified number by the method of Sieve of Eratosthenes.
6	The built-in data set mammals contain data on body weight versus brain weight. Develop R commands to:

a. Find the Pearson and Spearman correlation coefficients. Are they similar?
b. Plot the data using the plot command.
c. Plot the logarithm (log) of each variable and see if that makes a difference.

7. Develop R program to create a Data Frame with following details and do the following operations.

Item Code	Item Category	Item Price
1001	Electronics	700
1002	Desktop Supplies	300
1003	Office Supplies	350
1004	USB	400

a. Subset the Data frame and display the details of only those items whose price is greater than or equal to 350.
b. Subset the Data frame and display only the items where the category is either “Office Supplies” or “Desktop Supplies”
c. Create another Data Frame called “item-details” with three different fields item Code, Item Qty on Hand and Item Reorder Lvl and merge the two frames

8 Let us use the built-in dataset air quality which has Daily air quality measurements in New York, May to September 1973. Develop R program to generate histogram by using appropriate arguments for the following statements.

a. Assigning names, using the air quality data set.
b. Change colors of the Histogram
c. Remove Axis and Add labels to Histogram
d. Change Axis limits of a Histogram
e. Add Density curve to the histogram

9 Design a data frame in R for storing about 20 employee details. Create a CSV file named “input.csv” that defines all the required information about the employee such as id, name, salary, start_date, dept. Import into R and do the following analysis.

a. Find the total number rows & columns
b. Find the maximum salary
c. Retrieve the details of the employee with maximum salary
d. Retrieve all the employees working in the IT Department.
e. Retrieve the employees in the IT Department whose salary is greater than 20000 and write these details into another file “output.csv”

10 Using the built in dataset mtcars which is a popular dataset consisting of the design and fuel consumption patterns of 32 different automobiles. The data was extracted rom the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models). Format A data frame with 32 observations on 11 variables : [1] mpg Miles/(US) gallon, [2] cyl Number of cylinders [3] disp Displacement (cu.in.), [4] hp Gross horsepower [5] drat Rear axle ratio,[6] wt Weight (lb/1000) [7] qsec 1/4 mile time, [8] vs V/S, [9] am Transmission (0 = automatic, 1 = manual), [10] gear Number of forward gears, [11] carb Number of carburetors

Develop R program, to solve the following:

a. What is the total number of observations and variables in the dataset?
b. Find the car with the largest hp and the least hp using suitable functions
c. Plot histogram / density for each variable and determine whether continuous variables are normally distributed or not. If not, what is their skewness?

	d. What is the average difference of gross horse power(hp) between automobiles with 3 and 4 number of cylinders(cyl)? Also determine the difference in their standard deviations. e. Which pair of variables has the highest Pearson correlation?
11	Demonstrate the progression of salary with years of experience using a suitable data set (You can create your own dataset). Plot the graph visualizing the best fit line on the plot of the given data points. Plot a curve of Actual Values vs. Predicted values to show their correlation and performance of the model. Interpret the meaning of the slope and y-intercept of the line with respect to the given data. Implement using lm function. Save the graphs and coefficients in files. Attach the predicted values of salaries as a new column to the original data set and save the data as a new CSV file.

Reference Books

1. Cotton, R. (2013). Learning R: A Step-by-Step Function Guide to Data Analysis. 1st ed. O'Reilly Media Inc.
2. Jones, O., Maillardet. R. and Robinson, A. (2014). Introduction to Scientific Programming and Simulation Using R. Chapman & Hall/CRC, The R Series.
3. Davies, T.M. (2016) The Book of R: A First Course in Programming and Statistics. No Starch Press.

Course Outcomes

After completion of the course student will be able to

CO1: Explain the fundamental syntax of R data types, expressions and the usage of the R-Studio IDE

CO2: Develop a program in R with programming constructs: conditionals, looping and functions.

CO3: Apply the list and data frame structure of the R programming language.

C04: Use visualization packages and file handlers for data analysis.

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

COURSE CODE: 22UCS338L	**Project Management with Git**	**Credits: 01**
		Semester:3
L:T:P - **0:0:2**		CIEMarks:50
Total Hours/Week: **02**		SEEMarks:50

Course objectives:

- To familiar with basic command of Git
- To create and manage branches
- To understand how to collaborate and work with Remote Repositories
- To familiar with virion controlling commands
-

Practical Module

SL.NO	Experiments
1	Setting Up and Basic Commands Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.
2	Creating and Managing Branches Create a new branch named “feature-branch”. Switch to the “master” branch. Merge the “feature-branch” into “master”
3	Creating and Managing Branches Write the commands to stash your changes, switch branches, and then apply the stashed changes.
4	Collaboration and Remote Repositories Clone a remote Git repository to your local machine.
5	Collaboration and Remote Repositories Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.
6	Collaboration and Remote Repositories Write the command to merge “feature-branch” into “master” while providing a custom commit message for the merge.
7.	Git Tags and Releases Write the command to create a lightweight Git tag named “v1.0” for a commit in your local repository.
8	Advanced Git Operations Write the command to cherry-pick a range of commits from “source-branch” to the current branch.
9	Analysing and Changing Git History Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?
10	Analysing and Changing Git History Write the command to list all commits made by the author “JohnDoe” between “2023-01-01” and “2023-12-31”
11	Analysing and Changing Git History Write the command to display the last five commits in the repository.
12	Analysing and Changing Git History Write the command to undo the changes introduced by the commit with the ID “abc123”

Suggested Learning

1. Version Control with Git, 3rd Edition, by Prem Kumar Ponuthorai, Jon Loeliger Released October 2022, Publisher(s): O&#39;Reilly Media, Inc.
2. Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, https://git-scm.com/book/en/v2
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0130944433473699842782_shared/overview

Course Outcomes

After completion of the course student will be able to

1. Use the basics commands related to git repository
2. Create and manage the branches
3. Apply commands related to Collaboration and Remote Repositories
4. Use the commands related to Git Tags, Releases and advanced git operations
5. Analyse and change the git history

Course Outcomes	**Programme Outcomes (POs)**												**Program Specific Outcomes (PSOs)**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	2	1	1	-	3	2	1	1	-	3	-	-	**1**	-	-
CO2	1	2	2	2	3	1	2	2	2	3	-	-	**2**	-	-
CO3	1	1	1	1	3	1	1	1	1	3	-	-	**2**	-	-
CO4	1	1	1	1	1	1	1	1	1	1	-	-	**2**	-	-

Course Code: **22UCS348L**	**Data Visualization with Python**	Credits :01
		Semester:3
Hours/Week: (0: 0: 2: 0)		CIE Marks:50
Total Hours of pedagogy: 40		SEE Marks:50

Course Type: Practical

Course objectives:

- Implementation using Matplotlib for drawing different Plots.
- Demonstrate working with Seaborn, Bokeh.
- Working with Plotly for 3D, Time Series and Maps.

Sl. No.	**Experiments**
1	Practice on NumPy library
2	Practice on Pandas library
3	a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib. b) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib.
4	a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib. b) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib.
5	a) Write a Python program to illustrate Linear Plotting using Matplotlib. b) Write a Python program to illustrate liner plotting with line formatting using Matplotlib.
6	Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions.
7	a) Write a Python program to explain working with bokeh line graph using Annotations and Legends b) Write a Python program for plotting different types of plots using Bokeh.
8	Write a Python program to draw 3D Plots using Plotly Libraries.
9	a) Write a Python program to draw Time Series using Plotly Libraries. b) Write a Python program for creating Maps using Plotly Libraries.
10	a) Write a python program to visualize geographical data using geopandas library b) Write a python program to visualize geographical data in map using folium library

Python (Full Course): https://www.youtube.com/watch?v=_uQrJ0TkZlc

Pedagogy	For the above experiments the following pedagogy can be considered. Problem based learning, Active learning, MOOC, Chalk &Talk

Course outcomes (Course Skill Set):

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

CO1. Demonstrate the use of NumPy

CO2: Demonstrate use of Pandas

CO3. Use Matplotlib for drawing different Plots

CO4. Demonstrate working with Seaborn, Bokeh for visualization.

CO5. Use Plotly for drawing Time Series and Maps.

CO and PSO Mapping

Course Outcomes	**Programme Outcomes**												**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	1	1	1	1	1								3	0	0
CO2	1	1	1	1	1								3	0	0
CO3	2	3	2	1	1								3	1	3
CO4	2	3	3	3	1								3	3	3
CO5	3	2	3	2	1								2	1	3

National Service Scheme (NSS)		Semester	**3rd to 6th**
Course Code	**22UHS002M**	CIE Marks	**25*4 = 100**
Teaching Hours/Week (L:T:P: S)	0:0:3:1	SEE Marks	------
Total Hours of Pedagogy	40 hour Practical Session +15 hour Planning	Total Marks	**25*4 = 100**
Examination nature (SEE)	Activities Report Evaluation by College NSS Officer at the end of every semester (3rd to 6th semester)		
Credits	NCMC – Non-Credit Mandatory Course (Completion of the course shall be mandatory for the award of degree)		

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

1. Understand the community in general in which they work.
2. Identify the needs and problems of the community and involve them in problem –solving.
3. Develop among themselves a sense of social & civic responsibility & utilize their knowledgein finding practical solutions to individual and community problems.
4. Develop competence required for group-living and sharing of responsibilities & gain skills in mobilizing community participation to acquire leadership qualities and democratic attitudes.
5. Develop capacity to meet emergencies and natural disasters & practice national integrationand social harmony in general.

General Instructions - Pedagogy :

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

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

National Service Scheme (NSS) – Contents

1. Organic farming, Indian Agriculture (Past, Present and Future) Connectivity for marketing.
2. Waste management– Public, Private and Govt organization, 5 R’s.
3. Setting of the information imparting club for women leading to contribution in social andeconomic issues.
4. Water conservation techniques – Role of different stakeholders– Implementation.
5. Preparing an actionable business proposal for enhancing the village income and approach forimplementation.
6. Helping local schools to achieve good results and enhance their enrolment in Higher/ technical vocation.

7. Developing Sustainable Water management system for rural areas and implementation approaches.
8. Contribution to any national level initiative of Government of India. For eg. Digital India, Skill India, Swatch Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill development programs etc.
9. Spreading public awareness under rural outreach programs. (minimum5 programs).
10. Social connect and responsibilities.
11. Plantation and adoption of plants. Know your plants.
12. Organize National integration and social harmony events /workshops /seminars. (Minimum 02 programs).
13. Govt. school Rejuvenation and helping them to achieve good infrastructure.

NOTE:

- Student/s in individual or in a group Should select any one activity in the beginning of each semester till end of thatrespective semester for successful completion as per the instructions of NSS officer with the consent of HOD of thedepartment.
- At the end of every semester, activity report should be submitted for evaluation.

Distribution of Activities - Semester wise from 3rd to 6th semester

Sem	Topics / Activities to be Covered
3rd Sem for 25 Marks	1. Organic farming, Indian Agriculture (Past, Present and Future) Connectivity for marketing. 2. Waste management– Public, Private and Govt organization, 5 R's. 3. Setting of the information imparting club for women leading to contribution in social andeconomic issues.
4th Sem for 25 Marks	4. Water conservation techniques – Role of different stakeholders– Implementation. 5. Preparing an actionable business proposal for enhancing the village income and approach forimplementation. 6. Helping local schools to achieve good results and enhance their enrolment in Higher/ technical/ vocational education.
5th Sem for25 Marks	7. Developing Sustainable Water management system for rural areas and implementationapproaches. 8. Contribution to any national level initiative of Government of India. Foreg. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme, Skill developmentprograms etc. 9. Spreading public awareness under rural outreach programs.(minimum5 programs). 10. Social connect and responsibilities.
6th Sem for 25 Marks	11. Plantation and adoption of plants. Know your plants. 12. Organize National integration and social harmony events /workshops /seminars. (Minimum 02programs). 13. Govt. school Rejuvenation and helping them to achieve good infrastructure.

Pedagogy – Guidelines, it may differ depending on local resources available for the studyas well as environment and climatic differences, location and time of execution.

Sl No	Topic	Group size	Location	Activity execution	Reporting	Evaluation Of the Topic
1.	Organic farming, IndianAgriculture (Past, Present and Future) Connectivity for marketing.	May be individual or team	Farmers land/Villages/ roadside / community area/ College campus etc…..	Site selection /proper consultation/Continuous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
2.	Waste management– Public, Private and Govtorganization, 5 R's.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Site selection /proper consultation/Continuous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
3.	Setting of the information imparting club for women leading to contributionin social and economic issues.	May be individual or team	Women empowerment groups/ Consulting NGOs & Govt Teams /College campus etc…..	Group selection/proper consultation/Continu ous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
4.	Water conservation techniques – Role of different stakeholders– Implementation.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	site selection / proper consultation/Continu ous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
5.	Preparing an actionablebusiness proposal for enhancing the village income and approach for implementation.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Group selection/proper consultation/Continu ous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer

6.	Helping local schools toachieve good results and enhance their enrolment in Higher/ technical/ vocational education.	May be individual or team	Local government / private/ aided schools/Government Schemes officers/ etc…..	School selection/proper consultation/Continu ous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
7.	Developing SustainableWater management system for rural areas and implementation approaches.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	site selection/proper consultation/Continu ous monitoring/ Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
8.	Contribution to any national level initiative of Government of India.For eg. Digital India, Skill India, Swachh Bharat, Atmanirbhar Bharath, Make in India, Mudra scheme,Skill development programs etc.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Group selection/proper consultation/Continu ous monitoring / Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer

9.	Spreading public awareness under ruraloutreach programs.(minimum5 programs). ///// Social connect and responsibilities.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Group selection/proper consultation/Continu ous monitoring / Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
10.	Plantation and adoptionof plants. Know your plants.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Place selection/proper consultation/Continu ous monitoring / Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
11.	Organize National integration and socialharmony events /workshops /seminars. (Minimum02 programs).	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Place selection/proper consultation/Continu ous monitoring / Information board	Report should be submitted by individual to the concerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer
12.	Govt. school Rejuvenation andhelping them to achieve good infrastructure.	May be individual or team	Villages/ City Areas /Grama panchayat/ public associations/Governme nt Schemes officers/ campus etc…..	Place selection/proper consultation/Continu ous monitoring / Information board	Report should be submitted by individual to theoncerned evaluation authority	Evaluation as per the rubrics Of scheme and syllabus by NSSofficer

Plan of Action (Execution of Activities For Each Semester)

Sl.NO	Practice Session Description
1	**Lecture session by NSS Officer**
2	**Students Presentation on Topics**
3	**Presentation - 1 , Selection of topic, PHASE - 1**
4	**Commencement of activity and its progress - PHASE - 2**
5	**Execution of Activity**
6	**Execution of Activity**
7	**Execution of Activity**
8	**Execution of Activity**
9	**Execution of Activity**
10	**Case study based Assessment, Individual performance**
11	**Sector wise study and its consolidation**
12	**Video based seminar for 10 minutes by each student At the end of semester with Report.**

- In every semester from 3rd semester to 6th semester, Each student should do activities accordingto the scheme and syllabus.
- At the end of every semester student performance has to be evaluated by the NSS officer for theassigned activity progress and its completion.
- At last in 6th semester consolidated report of all activities from 3rd to 6th semester, compiledreport should be submitted as per the instructions.

Course outcomes (Course Skill Set):

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

CO1: Understand the importance of his / her responsibilities towards society.

CO2: Analyse the environmental and societal problems/issues and will be able to design solutions for the same.CO3: Evaluate the existing system and to propose practical solutions for the same for sustainable development. CO4: Implement government or self-driven projects effectively in the field.

CO5: Develop capacity to meet emergencies and natural disasters & practice national integration and social harmony in general.

Assessment Details for CIE (both CIE and SEE)

Weightage	**CIE – 100%**	• Implementation strategies of the project (NSSwork). • The last report should be signed by NSS Officer, the
Presentation - 1 Selection of topic, PHASE – 1	10 Marks	
Commencement of activity and its progress - PHASE – 2	10 Marks	

Case study based Assessment Individual performance	10 Marks	HOD and principal. • At last report should be evaluated by the NSSofficer of the institute. • Finally the consolidated marks sheet should besent to the university and also to be made available at LIC visit.
Sector wise study and its consolidation	10 Marks	
Video based seminar for 10 minutes by each student At the end of semester with Report.	10 Marks	
Total marks for the course in each semester	**50 Marks**	
Marks scored for 50 by the students should be Scale down to 25 marks In each semester for CIE entry in the VTU portal.		
25 marks CIE entry will be entered in University IA marks portal at the end of each semester 3rd to 6th sem, Report and assessment copy should be made available in the department semester wise.		
Students should present the progress of the activities as per the schedule in the prescribed practical session in the field.There should be positive progress in the vertical order for the benefit of society in general.		

Suggested Learning Resources:

Books :

1. **NSS Course Manual,** Published by NSS Cell, VTU Belagavi.
2. Government of Karnataka, NSS cell, activities reports and its manual.
3. Government of India, nss cell, Activities reports and its manual.

B.E. in Computer Science and Engineering
Scheme of Teaching and Examinations
Admitted Batch 2023-2024

IV SEMESTER

Sl. No	Course and Course Code		Course Title	Teaching Department (TD) and Question Paper Setting Board(PSB)	Teaching Hours/Week				Examination				Credits
					Theory Lecture	Tutorial	Practical/Drawing	Self-Study	Duration in hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1	PCC/ BS C	22UCS411C	Analysis & Design of Algorithms	TD: CSPSB:CS	3	0	0		03	50	50	100	3
2	IPCC	22UCS412C	Microcontrollers	TD: CS PSB:CS	3	0	2		03	50	50	100	4
3	IPCC	22UCS413C	Database Management Systems	TD: CS PSB:CS	3	0	2		03	50	50	100	4
4	PCCL	22UCS414L	Analysis &Design of Algorithms Lab	TD: CSPSB:CS	0	0	2		03	50	50	100	1
5	ESC	22UCS415C 22UCS425C 22UCS435C 22UCS445C	ESC/ETC/PLC	TD:CS/Maths PSB:CS/Maths	2	2	0		03	50	50	100	3
6	AEC/ SEC	22UCS416C 22UCS426C 22UCS436C 22UCS446L	Ability Enhancement Course/Skill Enhancement Course- IV	TD:Concerned department PSB:CS	If the course is Theory				01	50	50	100	1
					1	0	0						
					If the course is a lab				02				
					0	0	2						
4	BSC	BBOC407	Biology For Computer Engineers	TD /PSB: BT, CHE,	2	0	0		03	50	50	100	2

7	UHV	BUHK40 8	Universal human values course	Any Department	1	0	0		01	50	50	100	1
9	MC	BNSK459	National Service Scheme(NSS)	NSS coordinator	0	0	2			100	---	100	0
		BPEK459	Physical Education(PE) (Sports and Athletics)	Physical Education Director									
		BYOK459	Yoga	Yoga Teacher									
									Total	**500**	**400**	**900**	**19**

Course Code: 22UCS411C	**Analysis and Design of Algorithms**	Semester:4 Credits :03
Hours/Week: 03		CIE Marks: 50
Total Hours of padagogy: 40 Hrs		SEE Marks: 50
Course Type: Theory		

Course Objectives: ● To learn the methods for analyzing algorithms and evaluating their performance. ● To demonstrate the efficiency of algorithms using asymptotic notations. ● To solve problems using various algorithm design methods, including brute force, greedy, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking, and branch and bound. ● To learn the concepts of P and NP complexity classes.	
Module-1	**08 Hrs.**
INTRODUCTION: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving. **FUNDAMENTALS OF THE ANALYSIS OF ALGORITHM EFFICIENCY:** Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non recursive Algorithms, Mathematical Analysis of Recursive Algorithms. **BRUTE FORCE APPROACHES:** Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching. **Chapter 1 (Sections 1.1,1.2), Chapter 2(Sections 2.1,2.2,2.3,2.4),** **Chapter 3(Section 3.1,3.2)**	
Module-2	**08 Hrs.**
BRUTE FORCE APPROACHES (contd..): Exhaustive Search (Travelling Salesman problem and Knapsack Problem). **DECREASE-AND-CONQUER:** Insertion Sort, Topological Sorting. **DIVIDE AND CONQUER:** Merge Sort, Quick Sort, Binary Tree Traversals, Multiplication ofLarge Integers and Strassen's Matrix Multiplication. **Chapter 3(Section 3.4), Chapter 4 (Sections 4.1,4.2), Chapter 5 (Section 5.1,5.2,5.3, 5.4)**	
Module-3	**08 Hrs.**
TRANSFORM-AND-CONQUER: Balanced Search Trees, Heaps and Heapsort. **SPACE-TIME TRADEOFFS:** Sorting by Counting: Comparison counting sort, Input Enhancement in String Matching: Horspool's Algorithm. **Chapter 6 (Sections 6.3,6.4), Chapter 7 (Sections 7.1,7.2)**	
Module-4	**08 Hrs.**
DYNAMIC PROGRAMMING: Three basic examples, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. **THE GREEDY METHOD:** Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes. **Chapter 8 (Sections 8.1,8.2,8.4), Chapter 9 (Sections 9.1,9.2,9.3,9.4)**	
Module-5	**08 Hrs.**

LIMITATIONS OF ALGORITHMIC POWER: Decision Trees, P, NP, and NP-Complete Problems. **COPING WITH LIMITATIONS OF ALGORITHMIC POWER**: Backtracking (n-Queens problem,Subset-sum problem), Branch-and-Bound (Knapsack problem), Approximation algorithms for NP-Hard problems (Knapsack problem).

Chapter 11 (Section 11.2, 11.3), Chapter 12 (Sections 12.1,12.2,12.3)

Suggested Learning resources

Textbooks

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian),2017, Pearson.

Reference books

1. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, CliffordStein, 3rd Edition, PHI.
3. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

Web links and Video Lectures (e-Resources):

Design and Analysis of Algorithms: https://nptel.ac.in/courses/106/101/106101060/

Course Outcomes:

CO1: Apply asymptotic notational method to analyze the performance of the algorithms in terms oftime complexity.

CO2: Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems.

CO3: Make use of transform & conquer and dynamic programming design approaches to solve thegiven real world or complex computational problems.

CO4: Apply greedy and input enhancement methods to solve graph & string based computationalproblems.

CO5: Analyse various classes (P,NP and NP Complete) of problems

CO6 Illustrate backtracking, branch & bound and approximation methods.

CO and POs Mapping

Course Outcomes	Programme Outcomes												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	3	3	-	-	-	-	-	-	-	-	-	3	1	2
CO2	3	2	3	-	-	-	-	-	-	-	-	1	3	1	1
CO3	2	3	2	-	-	-	-	-	-	-	-	2	2	2	2
CO4	2	3	3	-	-	-	-	-	-	-	-	2	2	2	2
CO5	2	2	3	-	-	-	-	-	-	-	-	1	2	2	1
CO6	2	2	3	-	-	-	-	-	-	-	-	1	2	2	1

Semester: 4	**Microcontrollers**	**Course Code: 22CS412C**
Credits: 4		**LTP: 3:0:2**
No. of teaching Hours: **Lecture: 40 + Tutorial: 28**		**CIE Marks: 50**
Course Type: Integrated		**SEE Marks: 50**

Course Objectives:

1: Familiarize with ARM programming modules along with registers, CPSR and Flags.
2: Develop ALP using various instructions to program the ARM controller.
3: Understand the Exceptions and Interrupt handling mechanism in Microcontrollers.

Module-1	**8 Hrs.**

ARM Embedded Systems: The RISC design philosophy, The ARM Design Philosophy, EmbeddedSystem Hardware, Embedded System Software.

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

Module-2	**8 Hrs.**

Introduction to the ARM Instruction Set: Data Processing Instructions, Branch Instructions, Software Interrupt Instructions, Program Status Register Instructions, Coprocessor Instructions, Loading Constants

Module-3	**8 Hrs.**

C Compilers and Optimization: Basic C Data Types, C Looping Structures, Register Allocation, FunctionCalls, Pointer Aliasing, Portability Issues.

Module-4	**8 Hrs.**

Exception and Interrupt Handling: Exception handling, ARM processor exceptions and modes, vectortable, exception priorities, link register offsets, interrupts, assigning interrupts, interrupt latency, IRQ and FIQ exceptions, basic interrupt stack design and implementation.

Firmware: Firmware and bootloader, ARM firmware suite, Red Hat redboot, Example: sandstone, sandstonedirectory layout, sandstone code structure.

Module-5	**8 Hrs.**

CACHES: The Memory Hierarchy and Cache Memory, Caches and Memory Management Units: CACHE Architecture: Basic Architecture of a Cache Memory, Basic Operation of a Cache Controller, The Relationship between Cache and Main Memory, Set Associativity, Write Buffers, Measuring Cache Efficiency, CACHE POLICY: Write Policy—Writeback or Writethrough, Cache Line Replacement Policies, Allocation Policy on a Cache Miss. Coprocessor 15 and caches.

Practical Module

13. Using Keil software, observe the various Registers, Dump, CPSR, with a simple Assembly Language Programs (ALP).
14. Develop and simulate ARM ALP for Data Transfer, Arithmetic and Logical operations (Demonstrate with the help of a suitable program).
15. Develop an ALP to multiply two 16-bit binary numbers.
16. Develop an ALP to find the sum of first 10 integer numbers.
17. Develop an ALP to find the largest/smallest number in an array of 32 numbers.
18. Develop an ALP to count the number of ones and zeros in two consecutive memory locations.
19. Simulate a program in C for ARM microcontroller using KEIL to sort the numbers in ascending/descending order using bubble sort.

20. Simulate a program in C for ARM microcontroller to find factorial of a number.
21. Simulate a program in C for ARM microcontroller to demonstrate case conversion of charactersfrom upper to lowercase and lower to uppercase.
22. Demonstrate enabling and disabling of Interrupts in ARM.
23. Demonstrate the handling of divide by zero, Invalid Operation and Overflow exceptions in ARM.

Suggested Learning resources

Reference Books:

1. Andrew N Sloss, Dominic Symes and Chris Wright, ARM system developers guide, Elsevier, MorganKaufman publishers, 2008
2. Raghunandan G.H, Microcontroller (ARM) and Embedded System, Cengage learning Publication,2019.
3. Insider's Guide to the ARM7 based microcontrollers, Hitex Ltd.,1st edition, 2005

Course Outcomes:

1. Comprehend the ARM Architectural features and Instructions.
2. Develop programs using ARM instruction set for an ARM Microcontroller.
3. Analyze C-Compiler Optimizations and portability issues in ARM Microcontroller.
4. Apply the concepts of Exceptions and Interrupt handling mechanisms in developing applications.
5. Demonstrate the role of Cache management and Firmware in Microcontrollers.

Course Outcomes	**Programme Outcomes**												**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
CO1	1		3		2										
CO2			3	3											3
CO3	1	2											3		
CO4	1	2		3		2								3	
CO5		1										1			

COURSE CODE: 22CS413C	**DATABASE MANAGEMENT SYSTEMS**	**Credits: 04**
		Semester:4
L:T:P - **3:0:2:0**		CIE Marks: 50
Total Hours/Week: 40		SEE Marks: 50

Course objectives:

- To Provide a strong foundation in database concepts, technology, and practice.
- To Practice SQL programming through a variety of database problems.
- To Understand the relational database design principles.
- To Demonstrate the use of concurrency and transactions in database.
- To Design and build database applications for real world problems.
- To become familiar with database storage structures and access techniques.

Module-I 08 Hrs.

Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications.

Overview of Database Languages and Architectures: Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. **Conceptual Data Modelling using Entities and Relationships:** Entity types, Entity sets and structural constraints, Weak entity types, ER diagrams, Specialization and Generalization.

Textbook 1: Ch 1.1 to 1.8, 2.1 to 2.6, 3.1 to 3.10

RBT: L1, L2, L3

Module-II 08 Hrs.

Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations.

Relational Algebra: Unary and Binary relational operations, additional relational operations (aggregate, grouping, etc.) Examples of Queries in relational algebra.

Mapping Conceptual Design into a Logical Design: Relational Database Design using ER-to-Relational mapping.

Textbook 1: Ch 5.1 to 5.3, Ch 8.1 to 8.5; Ch 9.1 to 9.2 Textbook 2: 3.5

RBT: L1, L2, L3

Module-III 08 Hrs.

Normalization: Database Design Theory – Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, 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.

SQL: SQL data definition and data types, Schema change statements in SQL, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL

Textbook 1: Ch 14.1 to 14.7, Ch 6.1 to 6.5

RBT: L1, L2, L3

Module-IV 08 Hrs.

SQL: Advanced Queries: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL.

Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, characterizing schedules based on recoverability, characterizing schedules based on Serializability, Transaction support in SQL.

Textbook 1: Ch 7.1 to 7.3, Ch 20.1 to 20.6

RBT: L1, L2, L3

Module-V	08 Hrs.

Concurrency Control in Databases: Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking. **NOSQL Databases and Big Data Storage Systems:** Introduction to NOSQL Systems, The CAP Theorem, Document-Based NOSQL Systems and MongoDB, NOSQL Key-Value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j

Textbook 1: Chapter 21.1 to 21.5, Chapter 24.1 to 24.6

RBT: L1, L2, L3

Practical Module

Sl.NO	Experiments
1	Create a table called Employee & execute the following. **Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION)** 1. Create a user and grant all permissions to the user. 2. Insert the any three records in the employee table contains attributes EMPNO, ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.
2	Create a table called Employee that contain attributes EMPNO, ENAME, JOB, MGR, SAL & execute the following. 1. Add a column commission with domain to the Employeetable. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee whose Empno is 105.
3	Queries using aggregate functions (COUNT, AVG, MIN, MAX, SUM), Group by, Order by. **Employee (E_id, E_name, Age, Salary)** 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employee table 3. Find the Maximum age from employee table. 4. Find the Minimum age from employee table. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.
4	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. **CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)**
5	Create cursor for Employee table & extract the values from the table. Declare the variables, Open the cursor & extrct the values from the cursor. Close the cursor.

	Employee (E_id, E_name, Age, Salary)
6	Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.
7	Install an Open Source NoSQL Data base MangoDB & perform basic CRUD (Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

Reference Books

2. Ramez Elmasri and Shamkant B. Navathe, (2017), “Fundamentals of Database Systems”, 7th Edition, Pearson
3. Ramakrishnan, and Gehrke, (2014), “Database management systems”, 3rd Edition, McGraw Hill

Course Outcomes

After completion of the course student will be able to

1. Describe the basic elements of a relational database management system
2. Design entity relationship and develop database applications for the given scenario.
3. Apply various Structured Query Language (SQL) statements for database manipulation.
4. Analyse various normalization forms for the given application.
5. Explain the concepts related to NoSQL databases.

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

Course Code: 22UCS414L	**Analysis and Design of Algorithms Lab**	**Credits :01**
		Semester:4
Hours/Week: (02)		**CIE Marks : 50**
Total Hours gy: 2Hrs/Week		**SEE Marks : 50**

Course Objectives:

- To design and implement various algorithms in C/C++ programming using suitable development tools to address different computational challenges.
- To apply diverse design strategies for effective problem-solving.
- To Measure and compare the performance of different algorithms to determine their efficiency and suitability for specific tasks.

Sl.No	Experiments
1	Design and implement C/C++ Program to sort a given set of n integer elements using Selection Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2	Design and implement C/C++ Program to sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3	Design and implement C/C++ Program to sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
4	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.
5	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.
6	a. Design and implement C/C++ Program to solve All-Pairs Shortest Paths problem using Floyd's algorithm. b. Design and implement C/C++ Program to find the transitive closure using Warshall's algorithm.
7	Design and implement C/C++ Program to find shortest paths from a given vertex in a weighted connected graph to other vertices using Dijkstra's algorithm.
8	Design and implement C/C++ Program to obtain the Topological ordering of vertices in a given digraph.
9	Design and implement C/C++ Program to solve 0/1 Knapsack problem using Dynamic programming method.
10	Design and implement C/C++ Program to solve discrete Knapsack and continuous Knapsack problems using greedy approximation method.
11	Design and implement C/C++ Program to find a subset of a given set S = {sl , s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d.
12	Design and implement C/C++ Program for N Queen's problem using Backtracking.

Suggested Learning resources

Textbooks

1. Introduction to the Design and Analysis of Algorithms, By Anany Levitin, 3rd Edition (Indian), 2017, Pearson.

Reference books

4. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press.
5. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, CliffordStein, 3rd Edition, PHI.
6. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

Web links and Video Lectures (e-Resources):

Design and Analysis of Algorithms: https://nptel.ac.in/courses/106/101/106101060/

Course Outcomes:

CO1: Develop programs to solve computational problems using suitable algorithm design strategy.

CO2: Compare algorithm design strategies by developing equivalent programs and observing runningtimes for analysis (Empirical).

CO3: Make use of suitable integrated development tools to develop programs

CO4: Choose appropriate algorithm design techniques to develop solution to the computational andcomplex problems.

CO5: Demonstrate and present the development of program, its execution and running time(s) and record the results/inferences.

CO and POs Mapping

Course Outcomes	**Programme Outcomes**												**PSO**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**	**3**
CO1	3	-	3	3	-	-	-	-	-	-	-	3	3	2	-
CO2	3	-	3	3	-	-	-	-	-	-	-	1	2	2	-
CO3	1	2	-	-	-	-	-	-	-	-	-	2	2	1	-
CO4	2	3	3	-	-	-	-	-	-	-	-	2	1	1	-
CO5	2	3	3	-	-	-	-	-	-	-	-	1	1	1	-

Course Code:22UCS415C	**DISCRETE MATHEMATICAL STRUCTURES**	**Credits: 03** **Semester:4**
L:T:P:S – 2 : 2 : 0 : 0		CIE Marks: 50
Total Hours/Week: 04		SEE Marks: 50

Course objectives:

1. To help students to understand discrete and continuous mathematical structures.
2. To impart basics of relations and functions.
3. To facilitate students in applying principles of Recurrence Relations to find the generating functions and solve the Recurrence relations.
4. To have the knowledge of groups and their properties to understand the importance of algebraic properties relative to various number systems.

Module-I	**8 Hrs.**

Fundamentals of Logic: Basic Connectives and Truth Tables, Logic Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems.

Module-II	**8 Hrs.**

Properties of the Integers: Mathematical Induction, The Well Ordering Principle – Mathematical Induction, Recursive Definitions.

Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations – The Binomial Theorem, Combinations with Repetition.

Module-III	**8 Hrs.**

Relations and Functions: Cartesian Products and Relations, Functions – Plain and One-to-One, Onto Functions. The Pigeon-hole Principle, Function Composition and Inverse Functions.

Properties of Relations, Computer Recognition – Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams, Equivalence Relations and Partitions.

Module-IV	**8 Hrs.**

The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing isin its Right Place, Rook Polynomials.

Recurrence Relations: First Order Linear Recurrence Relation, The Second Order LinearHomogeneous Recurrence Relation with Constant Coefficients.

Module-V	**8 Hrs.**

Introduction to Groups Theory: Definitions and Examples of Particular Groups Klein 4-group, Additive group of Integers modulo n, Multiplicative group of Integers modulo-p and permutation groups, Properties of groups, Subgroups, cyclic groups, Cosets, Lagrange's Theorem.

Reference Books

1. **Ralph P. Grimaldi, B V Ramana,** 2004, **"Discrete Mathematical Structures an AppliedIntroduction",** 5th Edition, Pearson Education.
2. **Ralph P. Grimaldi,** 2004, **"Discrete and Combinatorial Mathematics"**, 5th Edition, PearsonEducation.
3. **Kenneth H. Rosen, 2007, "Discrete Mathematics and its Applications"**, 6th Edition, McGraw Hill.
4. **Jayant Ganguly,** 2010, **"A Treatise on Discrete Mathematical Structures",** Sanguine-Pearson.

5. **D.S. Malik and M.K. Sen, 2004, “Discrete Mathematical Structures Theory and Applications,** Latest Edition, Thomson.
6. **Thomas Koshy: Reprint 2008, “Discrete Mathematics with Applications”**, Elsevier, 2005.

Course Outcomes

After completion of the course, students will be able to

1. Apply logical reasoning concepts and mathematical proof techniques in proving theorems and statements.
2. Demonstrate the application of discrete structures in different fields of computer science.
3. Apply the basic concepts of relations, functions, and partially ordered sets for computer representations.
4. Solve problems involving recurrence relations and generating functions.
5. Illustrate the fundamental principles of Algebraic structures with the problems related to computer science & engineering.

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

Course Code: 22UCS425C	**GRAPH THEORY**	**Credits :**03
		Semester:4
Hours/Week: (2:2:0:0)		**CIE Marks :** 50
Total Hours of pedagogy: 40		**SEE Marks :** 50
Course Type: Theory		

Course Objectives: ● Understand the basic concepts of graphs and their properties, and operations of graphs. ● Hamiltonian and Euler graphs, trees and matrix representation of the graph. ● Apply the concepts of a planar graph, matching and coloring in computer science engineering.	
Module-1	**8 Hrs.**
Introduction to Graphs: Introduction- Basic definition – Application of graphs – finite, infinite andbipartite graphs – Incidence and Degree – Isolated vertex, pendant vertex and Null graph. Paths and circuits – Isomorphism, sub-graphs, walks, paths and circuits, connected graphs, disconnected graphs and components. **(RBT Levels: L1, L2 and L3)**	
Module-2	**8 Hrs.**
Eulerian and Hamiltonian graphs: Euler graphs, Operations on graphs, Hamiltonian paths and circuits, Travelling salesman problem. Directed graphs – types of digraphs, Digraphs and binary relation. **(RBT Levels: L1, L2 and L3)**	
Module-3	**8 Hrs.**
Trees – properties, pendant vertex, Distance and centres in a tree - Rooted and binary trees, counting trees, spanning trees. **Connectivity Graphs**: Vertex Connectivity, Edge Connectivity, Cut set and Cut Vertices, Fundamental circuits. **(RBT Levels: L1, L2 and L3)**	
Module-4	**8 Hrs.**
Planar Graphs: Planar graphs, Kuratowski's theorem (proof not required),Different representations of planar graphs, Euler's theorem, Geometric dual. Graph Representations: Matrix representation of graphs-Adjacency matrix, Incidence Matrix,Circuit Matrix, Path Matrix. **(RBT Levels: L1, L2 and L3)**	
Module-5	**8 Hrs.**
Graph Colouring: Colouring- Chromatic number, Chromatic polynomial, Matchings, Coverings, Four colour problem and Five colour problem. Greedy colouring algorithm. **(RBT Levels: L1, L2 and L3)**	
Suggested Learning resources	
Text Books: 1. Narsingh Deo, Graph theory with the applications to engineering & Computer Science, Dovers Publications, 2016 2. J.A. Bondy and U.S.R. Murty. Graph theory with Applications, Springer, 1st edition, 2008.	

Reference Books:

1. Garry Chartand and Ping Zhang, Introduction to Graph Theory, Tata McGraw-Hill, 2006.
2. Frank Harary, Graph Theory, Narosa Publishing House, Latest edition.
3. R. Diestel, Graph Theory, free online edition, 2016: diestel-graph-theory.com/basic.html.
4. Douglas B. West, Introduction to Graph Theory, Prentice Hall India Ltd.,2001, Robin J. Wilson, Introduction to Graph Theory, Longman Group Ltd.,2010

Web links and Video Lectures (e-Resources):

- **http://nptel.ac.in/courses.php?disciplineID=111**
- **http://www.class-central.com/subject/math(MOOCs)**
- **http://academicearth.org/**
- **VTU e-Shikshana Program**
- **VTU EDUSAT Program.**

Course Outcomes:

CO1: Explain the fundamental concepts of properties and representation of graphs.

CO2: Solve the problems involving characterization and operations on graphs.

CO3: Apply concepts of trees and graph connectivity to solve real world problems..

CO4: Apply the concepts of planar graph and graph representations to solve the given problem.

CO5: Use the concepts of matching and coloring of graphs to solve the real world problems

CO and PSO Mapping

Course Outcomes	Programme Outcomes												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	3	1	1	1									3	0	0
CO2	3	1	1	1									3	0	0
CO3	2	3	2	1									3	1	3
CO4	2	3	3	3									3	3	3
CO5	3	2	3	2									2	1	3

Course Code: 22UCS435C	**Optimization Technique**	**Credits :03**
		Semester:4
Hours/Week: (03)		**CIE Marks: 50**
Total Hours of padagogy: **40 Hrs**		**SEE Mark : 50**
Course Type: Theory		

Course Objectives:

The objectives of the course are to facilitate the learners to:

- Appreciate the importance of linear algebra in computer science and alliedengineering science.
- Gain the knowledge of linear algebra tools and concepts to implement them intheir core domain.
- Improve their mathematical thinking and acquire skills required for sustainedlifelong learning..

Module-1: Vector Calculus

08 Hrs.

Functions of several variables, Differentiation and partial differentials, gradients of vector-valued functions, gradients of matrices, useful identities for computing gradients, linearization and multivariate Taylor series.

(RBT Levels: L1, L2 and L3)

Module-2: Applications Of Vector Calculus

08 Hrs.

Backpropagation and automatic differentiation, gradients in a deep network, The Gradient of Quadratic Cost, Descending the Gradient of Cost, The Gradient of MeanSquared Error.

(RBT Levels: L1, L2 and L3)

Module-3: Convex Optimization-1

08 Hrs.

Local and global optima, convex sets and functions separating hyperplanes, application of Hessian matrix in optimization, Optimization using gradient descent, Sequential search 3- point search and Fibonacci search.

(RBT Levels: L1, L2 and L3)

Module-4: Convex Optimization-2

08 Hrs.

Unconstrained optimization -Method of steepest ascent/descent, NR method, Gradient descent, Mini batch gradient descent, Stochastic gradient descent.

(RBT Levels: L1, L2 and L3)

Module-5: Advanced Optimization

08 Hrs.

Momentum-based gradient descent methods: Adagrad, RMSprop and Adam.
Non-Convex Optimization: Convergence to Critical Points, Saddle-Point methods.

(RBT Levels: L1, L2 and L3)

Suggested Learning resources

Text Books:

1. Mathematics for Machine learning, Marc Peter Deisennroth, A. Aldo Faisal, ChengSoon Ong, 2020, Cambridge University Press.

2. S. Bubeck, Convex Optimization: Algorithms and Complexity, Foundations andTrends in Optimization, 2015.
3. S. Boyd, N. Parikh, and E. Chu,“ Distributed optimization and statistical learning viathe alternating direction method of multipliers”, Foundations and Trends in Machine Learning, Now Publishers Inc.

Reference Books:

1. Linear Algebra and Optimization for Machine Learning, Charu C. Aggarwal, Springer, 2020.
2. A. Beck, First-Order Methods in Optimization, MOS-SIAM Series on Optimization,2017.
3. F. Bach, “Learning with Submodular Functions: A Convex Optimization Perspective”, Foundations and Trends in Machine Learning, Now Publishers Inc.

Web links and Video Lectures (e-Resources):

- https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/index.htm
- https://www.math.ucdavis.edu/~linear/linear.pdf
- https://www.coursera.org/learn/linear-algebra-machine-learning
- https://nptel.ac.in/syllabus/111106051/
- https://github.com/epfml/OptML_course
- https://www.youtube.com/playlist?list=PL4O4bXkI-fAeYrsBqTUYn2xMjJAqlFQzX

Course Outcomes:

CO1: Apply the concepts of vector calculus to solve the given problem.

CO2: Apply the concepts of partial differentiation in machine learning and deep neural networks.

CO3: Analyze the convex optimization algorithms and their importance in computer science & engineering.

CO4: Apply the optimization algorithms to solve the problem.

CO5: Analyze the advanced optimization algorithms for machine learning

CO and POs Mapping

Course Outcomes	Programme Outcomes												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1															
CO2															
CO3															
CO4															
CO5															

Semester : 04	**LINEAR ALGEBRA**	**Sub. Code: 22UCS445C**
Credits : 03		**LTP: 3:0:0**
No. of teaching Hours:40		**CIE Marks: 50**
Course Type: Theory		**SEE Marks :50**

Course Objectives:

- To equip the students with standard concepts and tools in Linear algebra this will find them useful in their disciplines.
- Gain the knowledge of linear algebra tools and concepts to implement them in their core domain.
- Improve their mathematical thinking and acquire skills required for sustained lifelong learning.

Module-1	**8 Hrs.**

Introduction, Vector spaces, Subspaces, Linear Combinations, Linear Spans, row space and column space of a Matrix, Linear Dependence and Independence, Basis and Dimension, Coordinates.

Module-2	**8 Hrs.**

Introduction, Linear Mappings, Geometric linear transformation of i2, Kernel and Image of a linear transformations, Rank-Nullity Theorem (No proof), Matrix representation of linear transformations, Singular and Non-singular linear transformations, Invertible linear transformation

Module-3	**8 Hrs.**

Introduction, Polynomials of Matrices, Applications of Cayley-Hamilton Theorem, Eigen spaces of a linear transformation, Characteristic and Minimal Polynomials of Block Matrices, Jordan Canonical form.

Module-4	**8 Hrs.**

Inner products, inner product spaces, length and orthogonality, orthogonal sets and Bases, projections, Gram-Schmidt process, QR-factorization, least squares problem and least square error.

Module-5	**8 Hrs.**

Diagonalization and Orthogonal diagonalization of real symmetric matrices, quadratic forms and its classifications, Hessian Matrix, Method of steepest descent, Singular value decomposition. Dimensionality reduction – Principal component analysis.

Suggested Learning resources

Text Books:

1. David C. Lay, Steven R. Lay, Judi J Mc. Donald: “Linear Algebra and its applications”, Pearson Education, 6th Edition, 2021.
2. Gilbert Strang: “Linear Algebra and its applications”, Brooks Cole, 4th edition, 2005.

Reference Books:

1. Richard Bronson & Gabriel B. Costa: “Linear Algebra: An Introduction”, 2nd edition. Academic Press, 2014.
2. Seymour Lipschutz, Marc Lipso: “Theory and problems of linear algebra”, Schaum’s outline series - 6th edition, 2017, McGraw-Hill Education.
3. Marc Peter Deisennroth, A. Aldo Faisal, Cheng Soon Ong: “Mathematics for Machine learning”, Cambridge University Press, 2020.

Web links and Video Lectures (e-Resources):

- https://ocw.mit.edu/courses/mathematics/18-06sc-linear-algebra-fall- 2011/index.htm
- https://www.math.ucdavis.edu/~linear/linear.pdf
- https://www.coursera.org/learn/linear-algebra-machine-learning

- **https://nptel.ac.in/syllabus/111106051/**
- **http://nptel.ac.in/courses.php?disciplineID=111**
- **http://www.class-central.com/subject/math(MOOCs)**
- **http://academicearth.org/**
- **VTU e-Shikshana Program**
- **VTU EDUSAT Program.**

Course Outcomes:

1. Explain the concepts of vector spaces, subspaces, bases, dimension and their properties.
2. Use matrices and linear transformations to solve the given problem.
3. Compute Eigen values and Eigenvectors for the linear transformations
4. Determine orthogonality of inner product spaces.
5. Apply the optimization techniques to solve the problems.

CO, PO and PSO Mappings:

Course Outcomes	**Programme Outcomes**												**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
CO1	2	2	2	2									2		2
CO2	2	2	2	2									2		2
CO3	2	2	2	2									2		2
CO4	2	2	2	2									2		2
CO5	3	3	3	3									3		3

COURSE CODE: 22UCS416C	**Green IT and Sustainability**	**Credits: 01**
		Semester: 04
L**:T:P - NL : 1: 0:0**		CIE Marks: 50
Total Hours/Week: 01		SEE Marks: 50

Course objectives:

- Understand challenges for Green ICT and the environmental impact.
- Learn different aspects of ICT metrics and Sustainable Cloud Computing.
- Explore effects of software design on the sustainability.

Module-I 02 Hrs.

Green ICT -History, Agenda, and Challenges Ahead: Introduction, Industrial Revolution, The Emergence of Information and Communication Technologies, The Agenda and Challenges Ahead.

Module-II 03 Hrs.

Emerging Technologies and Their Environmental Impact: Introduction, Number of Connected Devices, Increased, Functionality, Increased Number of Separate Functions, Increased Demand for Speed and Reliability , Obsolescence—The Problem of Backward Compatibility, The Other Side of the Balance Sheet, Videoconference as an Alternative to Business Travel, Dematerialization of Product Chain, Travel Advice/Road Traffic Control, Intelligent Energy Metering , Building Management Systems, Saving IT

Module-III 03 Hrs.

Measurements and Sustainability: Introduction, ICT Technical Measures, Ecological Measures and Ethical Consideration, Systems Engineering for Designing Sustainable ICT-Based Architectures.

Module-IV 03 Hrs.

Sustainable Cloud Computing: Introduction, Challenges in the Use of Cloud Computing As Green Technology, Cloud Computing and Sustainability, Sustainable Applications of Cloud Computing, Technologies Associated With Sustainable Cloud Computing, Future Prospects of Sustainable Cloud Computing, Reflections on Sustainable Cloud Computing Applications.

Module-V 03 Hrs.

Sustainable Software Design: Overview and Scope, Evaluating Sustainability Effects , Sustainability and the Product Life Cycle , Direct Effects: Sustainability During Use, Runtime Energy Consumption Basics , Analyzing the Energy Consumption of an Application , Energy Consumption Reduction Using Physical Properties of Semiconductors, Optimizing the Energy Consumption of an Application: Compiler Techniques, Optimizing the Energy Consumption of an Application: Runtime Approaches.

Reference Books

1. Green Information Technology – A Sustainable Approach, Mohammad Dastbaz Colin Pattinson, BabakAkhgar, Elsevier, 2015 Inc
2. San Murugesan; G.R.Gangadharan, Harnessing Green IT: Principles and Practices, Wiley-IEEE Press.

Course Outcomes

After completion of the course student will be able to

1. Classify the challenges for Green ICT
2. Relate the environmental impact due to emerging technologies.
3. Demonstrate different aspects of ICT metrics.

4. Compare the various parameters related to Sustainable Cloud Computing.
5. Interpret the effects of software design on the sustainability.

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

SUBJECT CODE: 22UCS426C	Capacity Planning for IT	Credits: 01
L:T:P - NL : 1: 0: 0		CIE Marks: 50
Total Hours/Week: 14		SEE Marks: 50

Course objectives:

- Understand requirement and measurements for capacity planning, measurement and monitoring.
- Measurement of data for prediction towards the planning process.
- Understand concepts related to deployment, installation, configuration, and management.
- Role of virtualization and cloud services in capacity planning.

Module-I	03 Hrs.

Goals, Issues, and Processes: capacity planning, Quick and Dirty Math, Predicting When Your Systems Will Fail, Make Your System Stats Tell Stories, Buying Stuff: Procurement Is a Process, Performance and Capacity: Two Different Animals, The Effects of Social Websites and Open APIs.

Setting Goals for Capacity: Different Kinds of Requirements and Measurements, Architecture Decisions

Module-II	02 Hrs.

Measurement: Units of Capacity: Aspects of Capacity Tracking Tools, Applications of Monitoring.

Module-III	03 Hrs.

Measurement: API Usage and Its Effect on Capacity, Examples and Reality. **Predicting Trends:** Riding Your Waves.

Module-IV	03 Hrs.

Predicting Trends: Procurement, The Effects of Increasing Capacity, Long-Term Trends, Iteration and Calibration.

Deployment: Automated Deployment Philosophies, Automated Installation Tools, Automated Configuration.

Module-V	03 Hrs.

Virtualization and Cloud Computing: Virtualization, Cloud Computing, Computing Resource Evolutions, Mixed Definitions, Cloud Capacity, Use it or lose it (your wallet), Measuring the clouds, Cloud Case Studies, Cloud Use Case: Anonymous Desktop Software Company.

Reference Books

1. John Allspaw, The Art of Capacity Planning, 2008, O'Reilly

Course Outcomes

After completion of the course student will be able to.

1. Identify the requirement and measurements for capacity planning by considering the goal, issues, and processes.
2. Explain capacity measurement and monitoring.
3. Make use of measurement data for prediction towards overall planning process.
4. Explain the concepts related to deployment, installation, configuration, and management.
5. Demonstrate how the virtualization and cloud services fit into a capacity plan.

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

COURSE CODE: 22UCS436C	**UI/UX**	**Credits: 01**
		Semester:4
L:T:P - 1 : 0: 0		CIE Marks: 50
Total Hours/Week: 01		SEE Marks: 50

Course objectives:

- Understand user experience design requirements, with design goals, metrics and targets.
- Explore different prototyping methods, UX design principles with case examples.
- Understand the role of design thinking concepts and mental models in UX design.

Module-I	**03 Hrs.**

Introduction: Usability to user experience, Emotional impact as part of user experience, User experience needs a business case. Extracting Interaction Design Requirements: Needs & Requirements, Formal requirement extraction, Methods for requirement extraction.

Module-II	**03 Hrs.**

Design Thinking, Ideation, and Sketching: Design Thinking, Design Perspectives, User Personas, Ideation, Sketching. Mental Models and Conceptual Design: Storyboards, Design influencing user behaviour.

Module-III	**03 Hrs.**

Design Production: Detailed Design, Wireframes.
UX Goals, Metrics and Targets: UX Goals, UX Measures, Measurement instruments, UX Metrics.

Module-IV	**03 Hrs.**

Prototyping: Depth & breadth of a prototype, Fidelity of prototypes, Paper prototypes.
Connections with Software Engineering: Foundations for success in SE-UX development, The challenge of connecting SE and UX.

Module-V	**02 Hrs.**

UX Design Guidelines: Using and interpreting design guidelines, Human memory limitations, UX design guidelines & examples, Planning, Translation, Physical action, Outcomes, Assessment, Overall.

Reference Books

1.REX HARTSON and PARDHA S. PYLA, The UX Book-Process and Guidelines for Ensuring a Quality User Experience, Morgan Kaufmann, Elsevier, 2012.

Course Outcomes

After completion of the course student will be able to

1. Explain the user experience design requirements.
2. Relate design thinking concepts and mental models to UX design.
3. Illustrate UX design in line with design goals, metrics and targets.
4. Demonstrate different prototyping in relation with software engineering.
5. Explain UX design principles with case examples.

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

Semester: 4	**Technical Writing using LaTeX**	**Sub. Code: 22UCS446L**
Credits: 01		**L:T:P: 0:0:2**
No. of teaching Hours: **Lecture: 40 + Tutorials: 28**		**CIE Marks: 50**
Course Type: Laboratory		**SEE Marks: 50**

Course Objectives:

1: Apply Basic LaTeX Commands to Develop Simple Documents

2. Develop LaTeX Scripts to Present Tables and Figures in the Document

3. Illustrate LaTeX Scripts to Present Theorems and Mathematical Equations in the Document

4. Develop Programs to Generate a Complete Report with Citations and a Bibliography

5. Illustrate the Use of TikZ and Algorithm Libraries to Design Graphics and Algorithms in the Document

List Of Assignments

24. Develop a LaTeX script to create a simple document that consists of 2 sections [Section1, Section2], and a paragraph with dummy text in each section. And also include header [title of document] and footer [institute name, page number] in the document.
25. Develop a LaTeX script to create a document that displays the sample Abstract/Summary
26. Develop a LaTeX script to create a simple title page of the project Report [Use suitable Logos and text formatting]
27. Develop a LaTeX script to create the Certificate Page of the Report [Use suitable commands to leave the blank spaces for user entry]
28. Develop a LaTeX script to create a document that contains the following table with proper labels.

S.No	USN	Student Name	Marks Subject1	Marks Subject2	Marks Subject3

29. Develop a LaTeX script to include the side-by-side graphics/pictures/figures in the document by using the subgraph concept
30. Develop a LaTeX script to create a document that consists of the two mathematical equations.
31. Develop a LaTeX script to demonstrate the presentation of Numbered theorems, definitions, corollaries, and lemmas in the document
32. Develop a LaTeX script to demonstrate the presentation of lemmas in the document
33. Develop a LaTeX script to create a document that consists of two paragraphs with a minimum of 10 citations in it and display the reference in the section

Suggested Learning resources

Reference Books:

4. BOOK: A Short Introduction to LaTeX BY FIRUZA KARMALI (AIBARA), A book for beginners, 2019
5. BOOK: Formatting Information: A Beginner's Introduction to Typesetting with LaTeX, BY PETER FLYNN, Comprehensive TeX Archive Network (2005)
6. LaTeX TUTORIAL: [https://latex-tutorial.com/tutorials/]
7. LaTeX TUTORIAL: [https://www.javatpoint.com/latex]

Course Outcomes:

1. Apply basic LaTeX command to develop simple document
2. Develop LaTeX script to present the tables and figures in the document
3. Illustrate LaTeX script to present theorems and mathematical equations in the document
4. Develop programs to generate the complete report with citations and a bibliography
5. Illustrate the use of Tikz and algorithm libraries to design graphics and algorithms in the document

Course Outcomes	**Programme Outcomes**												**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**
CO1	1	3	2	-	2	1	-	-	-	-	-	-	-	-	**1**
CO2	2	2	3	-	2	1	-	-	-	-	-	-	**2**	-	**2**
CO3	1	2	2	-	3	-	-	-	1	-	-	-	-	**1**	**2**
CO4	2	1	1	-	2	1	-	-	1	-	-	**1**	-	**1**	-
CO5	1	2	1		1	2	-	-	--	-	-	**1**	**1**	-	**1**