MALNAD COLLEGE OF ENGINEERING, HASSAN

(An Autonomous Institution Affiliated to VTU, Belgaum)

Autonomous Programmes

Bachelor of Engineering

DEPARTMENT OF

INFORMATION SCIENCE AND ENGINEERING

SYLLABUS

III & IV Semester (2022 Admitted Batch)

(2nd Year)

Academic Year 2023-24

VISION

The department will be a premier centre focusing on knowledge dissemination and generation to address the emerging needs of information technology in diverse fields.

MISSION

1. To make students competent to contribute towards the development of IT field
2. Promote learning and practice of latest tools and technologies among students and prepare them for diverse career options
3. Collaborate with industry and institutes of higher learning for Research and Development, innovations and continuing education
4. Developing capacity of teachers in terms of their teaching and research abilities
5. Develop software applications to solve engineering and societal problems

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will be successful professionals in IT industry with good design, coding and testing skills, capable of assimilating new information and solve new problems

PEO2: Graduates will communicate proficiently and collaborate successfully with peers, colleagues and organizations

PEO3: Graduates will be ethical and responsible members of the computing profession and society

PEO4: Graduates will acquire necessary skills for research, higher studies, entrepreneurship and continued learning to adopt and create new applications

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

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

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

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

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

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

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

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

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

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

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

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

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

PROGRAM SPECIFIC OUTCOMES (PSOs)

Upon graduation, students with a degree B.E. in Information Science & Engineering will be able to:

1. Design and Develop efficient information systems for organizational needs.
2. Ability to adopt software engineering principles and work with various standards of Computing Systems.

Scheme of Evaluation (Theory Courses)

Assessment	Marks
THREE CIEs conducted for a total of 30 marks	30
Activities as decided by course faculty	20
SEE	50
Total	100

Scheme of Evaluation (Laboratory Courses)

Assessment	Marks
Continuous Evaluation in every lab session by the Course Coordinator	**10**
Record Writing	**20**
Laboratory CIE conducted by the Course Coordinator	**20**
SEE	**50**
Total	**100**

Examination	Maximum Marks	Minimum marks to qualify
CIE	**50**	**20**
SEE	**50**	**20**

Scheme & Syllabus for II Year
B. E. Information Science and Engineering
Academic Year 2023-24

Third Semester					
Course Category	**Course Code**	**Course Title**	**L-T-P in hours**	**Credits**	**Contact Hours**
BSC	22MA301	Mathematics for Computer Science	3-2-0	4	5
PCC	22IS302	Digital Design & Computer Organization	3-0-2	4	5
PCC	22IS303	Operating Systems	3-2-0	4	5
PCC	22IS304	Data Structures and Applications	3-0-2	4	5
PCCL	22IS305	Unix Lab	0-0-2	1	2
ESC	22IS306	ESC/ETC/PLC	2-0-2	3	4
UHV	22SCR	Social Connect and Responsibility	0-0-2	1	2
AEC / SEC	22IS307	Ability Enhancement Course/Skill Enhancement Course – III	0-0-2	1	2
Total				22	30

Engineering Science Course (ESC/ETC/PLC)			
22IS306A	OOP with Java	22IS306C	Discrete Mathematical Structures
22IS306B	OOP with C++	22IS306D	Graph Theory and Combinatorics
Ability Enhancement Course – III			
22IS307A	R Programming	22IS307C	Data Visualization with Python
22IS307B	Data Analytics with Excel	22IS307D	Version Controller with GiT

Fourth Semester					
Course Category	**Course Code**	**Course Title**	**L-T-P in hours**	**Credits**	**Contact Hours**
PCC	22IS401	Software Engineering	2-2-0	3	4
PCC	22IS402	Microprocessor and Microcontroller	2-0-2	3	4
PCC	22IS403	Database Management Systems	3-0-2	4	5
PCC	22IS404	Design and Analysis of Algorithms	3-0-2	4	5
ESC	22IS405	ESC/ETC/PLC	2-0-2	3	4
AEC/SEC	22IS406	Ability Enhancement Course/Skill Enhancement Course- IV	1-0-0	1	1
BSC	22BEIS407	Biology For Engineers	0-2-0	1	2
UHV	22UHV	Universal Human Values	0-2-0	1	2
Total				20	27

Engineering Science Course (ESC/ETC/PLC)			
22IS405A	C# and .Net Technologies	22IS405C	Optimization Techniques
22IS405B	Internet of Things	22IS405D	Probabiltiy, Stastistics and queing
Ability Enhancement Course – IV			
22IS406A	Green IT and Sustainability	22IS406C	Introduction to Web Technology
22IS406B	User Interface Design	22IS406D	Technical writing using Latex

Course Title	**DIGITAL DESIGN AND COMPUTER ORGANIZATION**		
Course Code	**22IS302**	**(L-T-P)C**	**(3-0-2)4**
Exam	**3 Hrs**	**Hours/Week**	**5**
SEE	**50 Marks**	**Total Hours**	**50 (36L+14P)**

Course Objective: Students will be able to design synchronous and asynchronous circuits.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Describe the behaviour of basic components of Digital circuits &Apply different methods for simplifying Boolean equations and design circuits	1	-
2	Analyse the working/Functionality of Different data processing circuits. Carryout the task of designing efficient combinational and sequential logic circuit implementations from functional description of digital systems.	1,2	2
3	Understand the instruction execution mechanism of a computer, working of processing unit, input/output operations.	1	
4	Analyze different memory organizations and cache mapping policies.	1	

MODULE – 1 **9 Hrs.**

Digital Logic: Overview of Basic Gates and Universal Logic Gates, AND-OR-Invert Gates, Positive and Negative Logic. Combinational Logic Circuits: Boolean Laws and Theorems, Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs, Quads, and Octets, Karnaugh Simplifications, Don’t Care Conditions, Product-of-Sums method, Sum-of-Products method Simplification by Quine-McCluskey Method.

Data-Processing Circuits: Multiplexers, Demultiplexers, EX-OR gates, Parity Generators and Checkers,

MODULE – 2 **9 Hrs.**

Data-Processing Circuits (contd..): Magnitude Comparator **Arithmetic Circuits**: Arithmetic Building Blocks, **Flip-Flops**: RS Flip-Flops, Gated Flip-Flops, Edge-triggered RS, D, JK Flip-Flops, JK Master-Slave Flip-Flops, Various Representations of Flip-Flops, **Registers**: Registers: Types of Registers, Serial In-Serial Out, Serial In-Parallel Out, Parallel In-Serial Out, Parallel In-Parallel Out.

MODULE -3 **9 Hrs.**

Basic Structure of Computers: Basic Operational Concepts**,** Numbers, Arithmetic Operations and Characters, Memory Location and Addresses**.**

Input/Output Organization: Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Direct Memory Access, Buses, Standard I/O Interfaces – PCI Bus, USB.

MODULE -4 **9 Hrs.**

Memory System: Basic Concepts, Semiconductor RAM Memories (till asynchronous DRAMs), Read Only Memories, Speed, Size and Cost, Cache Memories – Mapping Functions.

Arithmetic: Multiplication of positive numbers, Signed operand multiplication, Fast multiplication – bit-pair recoding of multipliers**,** Integer Division.

Basic Processing Unit: Some Fundamental Concepts, Execution of a Complete Instruction, Multiple Bus Organization.

Practical Component

<u>Guided Experiments :</u>

1. Realize the behaviour of following gates:
   i) AND ii) OR 3) NOT 4) NOR 5) NAND
2. Design and implement BCD to Excess-3 code converter.
3. Design and implement a Johnson Counter using 4-bit Shift Register IC.
4. Design and implement an Asynchronous Counter using 4 bit Binary Counter IC to count up from 0 to (n<15).

5. Design and implement a Full Adder using 4:1 multiplexer chip.
6. Simplify and realize given Boolean expressions using logic gates (y = C'+A'D'+B'D').

Open ended Experiments :

1. Design and implement: i) 4:1 Multiplexer ii) 2:4 Decoder using NAND gates.
2. Design and implement a 3 stage Asynchronous Counter using J-K flip flops to count down from 7 to n .
3. Given any 4-variable logic expression (y = C'+A'D'+B'D'), simplify using a Karnaugh Map and realize the simplified logic expression using 8:1 Multiplexer IC.
4. Realize Full Subtractor using only NAND gates.
5. Realize Full Adder using only Nand gates.

Text Books:

1. Donald P Leach, Albert Paul Malvino and Goutam Saha, Digital Principles and Applications, 8th Edition, McGraw Hill, 2017
2. Carl Hamacher, Zvonko Vranesic Sfwat Zaky: "Computer Organization", 5th Edition, Tata McGraw Hill 2002

Reference Books:

1. A. Anand Kumar, Digital Electronics, PHI
2. R.P Jain, Modern Digital Electronics, TMH
3. Stephen Brown, ZvonkoRanesic , Fundamentals of Digital Logic with verilog Design, TMH,2006 1.
4. Charles H. Roth, Jr, Fundamentals of Logic Design, , 5th Edition, Thomson, 2004
5. William Stallings: "Computer Organization & Architecture", 7th Edition, PHI, 2006
6. David A Patterson, John L Hennessy, "Computer Organization and Design"- The Hardware/software Interface ARM Edition, 4th Edition, Elsevier,2009

MOOC Course:

1. Switching circuits and logic design https://nptel.ac.in/courses/106/105/106105185/
1. Digital Circuits and Systems SWAYAM IIT-Madras , https://swayam.gov.in/ndl_noc19_ee51

Course Outcomes	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2	3	3												
CO3	3													
CO4	3													

Course Title	**OPERATING SYSTEMS**		
Course Code	**22IS303**	**(L-T-P)C**	**(3-2-0)4**
Exam	**3 Hrs**	**Hours/Week**	**5**
SEE	**50 Marks**	**Total Hours**	**50**

Course Objective: Students will be able to design synchronous and asynchronous circuits.
Course outcomes: At the end of course, student will be able to:

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Describe various concepts and functionalities of operating systems	1,2	-
2	Apply different Process scheduling , Disk Scheduling and memory management Algorithms	3	-
3	Apply different mechanisms for handling deadlocks and synchronization problems	3	
4	Implement various CPU scheduling algorithms , disk scheduling , page replacement algorithms	2,3	-

MODULE – 1	**9 Hrs.**

Introduction to Operating Systems, System Structures: What Operating Systems Do? Computer System Architecture; Operating System Structure; Operating System Operations; Operating System Services; System Calls; Types of System Calls; System Programs, **Process Management:** Process Concept; Operations on Processes; Inter-Process Communication. Multi-Threaded Programming: Overview; Multithreading Models.

MODULE – 2	**9 Hrs.**

Process Management (contd..): Process Scheduling: Basic Concepts; Scheduling Criteria; Scheduling Algorithms; **Process Synchronization**: The Critical Section Problem; Peterson's Solution; Synchronization Hardware; Semaphores; Classic problems of Synchronization, Monitors- Usage, Dining-Philosophers solution using monitors.

MODULE -3	**9 Hrs.**

Deadlocks: System Model; Deadlock Characterization; Methods for Handling Deadlocks; Deadlock Prevention; Deadlock Avoidance; Deadlock Detection and Recovery from Deadlock. **Memory Management:** Memory Management Strategies: Background; Swapping; Contiguous Memory Allocation; Paging; Structure of Page Table; Segmentation.

MODULE -4	**9 Hrs.**

Virtual Memory Management: Background; Demand Paging; Page Replacement. **Storage Management:** Secondary Storage Structures, Protection: Mass Storage Structures; Disk Structure; Disk Scheduling; Swap Space Management. **Protection:** Goals of Protection, Principles of Protection, Domain of Protection- Domain Structure, Access Matrix, Implementation of Access Matrix.

Text Books:

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Concepts, 9th Edition, 2018, John Wiley & Sons, ISBN 978-1-265-5427-0

Reference Books:

1. Andrew.S.Tanenbaum: Modern Operating Systems - A Concept Based Approach, 4th Edition, Addison Wesley, 2015. ISBN: 978-0133591620.
2. P.C.P. Bhatt: Introduction to Operating Systems: Concepts and Practice, 2nd Edition, PHI, 2008.
3. William Stallings: Operating Systems Internals and Design Principles, 8th Edition, Tata McGraw-Hill Education, 2007, ISBN - 978-0070611948.

MOOC Course:

1. Fundamentals of Operating System https://nptel.ac.in/courses/106/105/106105214/

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3												
CO2			3											
CO3			3											
CO4		3	3											

Course Title	DATA STRUCTURES AND APPLICATIONS		
Course Code	22IS304	L-T-P	(3-0-2) 4
Exam	3 Hrs.	Hours/Week	5
SEE	50 Marks	Total Hours	50 (36L+14P)

Course Objective: Students will be able to use appropriate data structures for solving problems.
Course outcomes: At the end of course, student will be able to:

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Describe various operations on data structures like Arrays, Lists, Stacks, Queues and Trees.	1	-
2.	Apply linear and non-linear data structures for solving problems.	2	1
3.	Design solutions for problems using appropriate data structures.	2	1
4.	Develop programs to solve a problem using data structures like stack, queue, list and tree.	2	2

MODULE – 1 — **9 Hrs.**

Introduction to Data Structures: Definition, Classification of Data Structures, Dynamic Memory Allocation – Introduction, Dynamic memory allocation, malloc, calloc, free and realloc. The Stack - Definition and examples: Primitive operations, Example. Representing stacks in C: Implementing the pop operation, testing for exceptional conditions, implementing the push operation. **Infix, postfix and prefix:** Basic definitions and examples, evaluating a postfix expression, Program to evaluate a postfix expression, converting an expression from infix to postfix, Program to convert an expression from infix to postfix.

MODULE – 2 — **9 Hrs.**

Recursion: Recursive definition and processes: Factorial function, Multiplication of natural numbers, Fibonacci sequence, Binary search, Properties of recursive definition or algorithm. **Recursion in C:** Factorial of a number, generation of Fibonacci numbers, Binary searching, Concept of Recursive chains, Towers of Hanoi problem, **Queues and lists:** The queue and its sequential representation: C implementation of queues, Insert operation, Priority queue, Array implementation of a priority queue. **Linked lists:** Inserting and removing nodes from a list, Linked implementation of stacks. Getnode and freenode operations.

MODULE -3 — **9 Hrs.**

Lists in C: Array implementation of lists, Limitations of array implementation, allocating and freeing dynamic variables, linked lists using dynamic variable, Queues as lists in C, Examples of list operations in C, Non integer and non-homogeneous lists. **Other list structures:** Circular lists, Stack as a circular list, Queue as a circular list.

MODULE -4 — **9 Hrs.**

Trees - Binary trees: Operations on binary trees, Applications of binary trees. Binary tree representation: Node representation of binary tree, Internal and external node, Implicit array representation of binary trees, choosing a binary tree representation, Binary tree traversals in C, Threaded binary trees.

Practical Component:

1. Design and Implement a menu driven Program in C for the following Array operations:
   a. Creating an Array of N Integer Elements
   b. Display of Array Elements with Suitable Headings
   c. Inserting an Element (ELEM) at a given valid Position (POS)
   Deleting an Element at a given valid Position (POS)

2. Write a C Program to create a Sequential file with at least 5 records, each record having the structure shown below:

USN	Name	Marks1	Marks2	Marks3
Non-zero positive integer	25 characters	Positive integer	Positive integer	Positive integer

Write necessary functions

a. To display all the records in the file.

To search for a specific record based on the USN.

3. Write a menu driven C Program to arrange a pile of dinner plates that you encounter when you eat at the local cafeteria: When you remove a plate from the pile, you take the plate on the top of the pile. This is exactly the plate that was added most recently to the pile by the dishwasher. If you want the plate at the bottom of the pile, you must remove all the plates on top of it to reach it (use integers to number dinner plates).

4. Write recursive C Programs for

a. Searching an element in a given list of integers using the Binary search method.

b. Solving the Towers of Hanoi problem.

5. Write a C Program to evaluate a valid suffix/postfix expression using stack. Assume that the suffix/ postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).

6. Write a menu driven C Program to simulate the working of a queue of vehicles on toll-tax bridge: The vehicle that comes first to the toll tax booth leaves the booth first. The vehicle that comes last leaves last. Therefore, it follows first-in-first-out (FIFO) strategy of queue (use integers to represent vehicles).

7. Write a menu driven C Program to simulate the working of a Circular Queue of integers using an array. Provide the following operations:

a. Insert b. Delete c. Display

8. Write a menu driven C Program using dynamic variables and pointers, to construct a Singly linked list of integers and perform insertion and deletion operations.

9. Write a menu driven C Program using dynamic variables and pointers to construct a Stack of integers using Singly linked list and to perform the following operations:

a. Push b. Pop c. Display

10. Write a menu driven C Program

a. To construct a binary search tree of integers.

b. To traverse the tree using all the methods i.e., In-order, Pre-order and Post-order.

Text Books:

1. Yedidyah Langsam and Moshe J. Augenstein and Aaron M.Tenanbaum, Data structures using C and C++, PHI, 2006 Chapters 2, 3, 4, 5

Reference Books:

1. Data Structures: A Pseudo-code approach with C –Gilberg and Forouzan, 2nd edition, Cengage Learning, 2014.
2. M.G.Venkateshmurthy, Programming techniques through C - A beginner’s companion, Pearson Education, Asia.
3. An Introduction to Data Structures with Applications- Jean-Paul Tremblay & Paul G. Sorenson, 2ndEdition, McGraw Hill, 2013.

MOOC Course:

2. Data Structures and algorithms https://nptel.ac.in/courses/106/102/106102064/

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	**3**													
CO2		**3**											**2**	
CO3		**3**											**2**	
CO4		**2**											**2**	

Course Title	UNIX LABORATORY		
Course Code	22IS305	L-T-P	(0-0-2)1
Exam	3 Hrs.	Hours/Week	2
SEE	50 Marks	Total Hours	28

Course Objective: The course provides a comprehensive introduction to UNIX user commands and utilities and students will develop Shell Programming and Vi editing skills.

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

#	Course Outcomes	Mappingto POs	Mappingto PSOs
1	Execute the basic UNIX commands.	2	-
2	Demonstrate and develop UNIX shell scripts	3	-

Execution following basic UNIX commands

ls ,ls –ld , ls –d, cat, cd ,od , mkdir,echo ,date , mv , cal ,wc,pwd,rmdir,touch,sort ,read, clear,ps ,whoami, alias , find ,test,expr,set ,shift ,type .

Design and Develop a shell scripts for following statements

1. Write a shell script that takes a valid directory name as an argument and recursively descend all the sub-directories, finds the maximum length of any file in that hierarchy and writes this Maximum value to the standard output.
2. Write a shell script that accepts a path name and creates all the components in that path name as directories. For example, if the script is named mpc, then the command mpc a/b/c/d should create directories a, a/b, a/b/c, a/b/c/d.
3. Write a shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each file name followed by its permissions.
4. Create a script file called file-properties that reads a file name entered and outputs it properties.
5. Write a shell script that accept one or more filenames as argument and convert all of them to uppercase, provided they exist in current directory
6. Write a shell script that accepts as filename as argument and display its creation time if file exist and if it does not send output error message.
7. Write a shell script that gets executed displays the message either “Good Morning” or “Good Afternoon” or “Good Evening” depending upon time at which the user logs in.
8. Write a shell script that accept the file name, starting and ending line number as an argument and Display all the lines between the given line number.

course Articulation Matrix

Course Outcomes	Program Outcomes [POs]												
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1		3												
CO2			2											

Course Title	OBJECT ORIENTED PROGRAMMING WITH JAVA

Course Code	**22IS306A**	**L-T-P-C**	**(2-0-2)3**
Exam	**3 Hrs**	**Hours/Week**	**4**
SEE	**50 Marks**	**Total Hours**	**40 (26L + 14P)**

Course Objective: Students will be able to apply Object Oriented Programming concepts for designing Applications

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

#	**Course Outcomes**	**Mapping to POs**	**Mapping to PSOs**
1	Comprehend the fundamental concepts Object Oriented Programming	1, 2	1
2	Apply Object Oriented constructs for program development	2	1
4	Analyze a java program for identifying bugs	2, 3	1

MODULE-1	**7 Hrs**

Object Oriented Concepts and Java: Concepts of Object-Oriented programming language: Object, Class, Message passing, inheritance, encapsulation, and polymorphism Difference between OOP and other conventional programming – advantages and disadvantages of OOP. Introduction to Java: Java and Java Applications, Java Development Kit(JDK), The Byte Code, The Java Buzzwords, Simple Java Programs using Control Statements and Blocks of code, Lexical Issues. Data Types, Variables, and Arrays The primitive Types, Integers, Floating-Point Types, Characters, Booleans, Variables, Type conversion and Casting, Arrays, Strings.

MODULE-2	**6 Hrs.**

Operators: Arithmetic, Bitwise, Relational, Boolean Logical, Assignment Operator, The '?' Operator, Operator Precedence. Program Control Statements: Input characters from the Keyword, if statement, Nested ifs, if-else-if Ladder, Switch Statement, Nested switch statements, for Loop, Enhanced for Loop, While Loop, do-while Loop, use break, Use continue, Nested Loops.

MODULE-3	**6Hrs**

Introducing Classes, Objects and Methods: Class Fundamentals, Declaring Objects, Object Reference Variables, Methods, Constructors, the "This" keyword, Garbage collection, Overloading Methods, and constructors, Argument Passing, Returning Objects, Access Control, Nested and Inner Classes.

MODULE-4	**7Hrs**

Inheritance, Packages, and Interfaces: Inheritance Basics, Using Super, Multilevel Hierarchy, When Constructors are called, Method Overriding, Abstract Classes. Packages, Access Protection, Importing Packages, Interfaces

Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and Catch, Multiple catch Clauses, throw, Java's Built-in Exceptions, Customized exceptions.

Text Books:

1. Herbert Schildt, "Java the Complete Reference, 9th Edition, Tata McGraw Hill
2. E- Bala Guruswamy 7th edition 2021

Reference Books:

1. Y. Daniel Liang, "Introduction to JAVA Programming, Brief Version", 9th Edition, Pearson Education,2019

Practical component

1. Write a Java Program that grades multiple-choice tests. Suppose there are 'm' students and 'n' Questions, and the answers are stored in a two-dimensional array. Each row records a student's answers to the questions. The answer key is stored in a one-dimensional array. The program grades the test and displays the result.
2. Write a Java program to count the number of occurrences of each letter in a string regardless of case.
3. Write a Java program that ignores non-alphanumeric characters in checking whether a string is a palindrome.
4. Create a java program to make a calculator using switch case to do arithmetic operations.
5. Create a Class Vehicle. Demonstrate method overriding by considering any two types of vehicles.
6. Create a super class called Figure that stores the dimensions of a two-dimensional object. Using Dynamic polymorphism in Java, compute the area of any three two-dimensional figures.
7. Write a Java program to demonstrate use of interfaces for computing the Net balance Amount after considering Gross income and expenditures in an Employee of an organization.
8. Write a program to demonstrate use of extending interfaces.
9. Write a java program to demonstrate handling of Array Index Out Of Bounds Exception and Arithmetic Exception.

TextBooks:

1. Herbert Schildt, "Java the Complete Reference, 9th Edition, Tata McGraw Hill.
2. E- Balaguruswamy 7th edition 2021.

Reference Books:

1. Y. Daniel Liang, "Introduction to JAVA Programming, Brief Version", 9th Edition, Pearson Education, 2019.

MOOC:

https://onlinecourses.nptel.ac.in/noc22_cs47/preview

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3											2	
CO2		3											2	
CO3		3	3										2	

Course Title	**OBJECT ORIENTED PROGRAMMING WITH C++**		
Course Code	**22IS306B**	**(L-T-P)C**	**(2-0-2)3**

Exam	**3Hrs**	**Hours/Week**	**4**
SEE	**50 Marks**	**Total Hours**	**40 (26L+14P)**

Course Objective : Students will be able to apply object oriented programming concepts in development of applications.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Describe all concepts of OOPs	1	
2.	Conduct experiments to demonstrate OOPs concepts	2	1
3.	Develop solutions to problems using OOPs principles	2	1

MODULE – 1 **7 Hrs.**

Principles of Object Oriented Programming: Object Oriented Programming Paradigm; Basic concepts of Object Oriented Programming; A Simple C++ Program; More C++ Statements; An Example with Class; Structure of C++ Program; Creating the Source File; Compiling and linking.Functions in C++: Introduction; The Main Function; Function Prototyping; Call by Reference; Return by Reference; Inline Function; Default Arguments; Const Arguments; Functions Overloading;

MODULE – 2 **7 Hrs.**

Classes and Objects: Specifying a Class; Defining Member Functions; A C++ Program with Class; Making an Outside Function Inline; Nesting of Member Functions; Memory Allocation for Objects; Static Data Members; Static Member Functions; Array of Objects; Objects as Function Arguments; Constructors and Destructors: Introduction; Constructors; Parameterized Constructors; Destructors.

MODULE -3 **6 Hrs.**

Operator Overloading and Type Conversions: Introduction; Defining Operator Overloading; Overloading Unary Operators; Overloading Binary Operators; Manipulation of String Using Operators; Rules for Overloading Operators; Inheritance; Extending Classes;Introduction; Defining derived Classes; Single Inheritance; Making a Private Member Inheritable; Multilevel Inheritance; Hierarchical Inheritance; Hybrid Inheritance;

MODULE -4 **6 Hrs.**

Templates: Introduction; Class Templates; Class Templates with Multiple Parameters; Function Templates; Function Templates with Multiple Parameters; Exception Handling: Introduction; Basic of Exception Handling; Exception Handling Mechanism; Throwing Mechanism; Catching Mechanism; Rethrowing an Exception; Specifying Exceptions.

Practical Component

1. Implement a simple C++ program to create a class with two data members and three member functions, create objects of this class and call the functions.
2. Using reference variables and inline functions, Implement a C++ program to find the average of three real numbers.
3. Using function overloading concept, Implement a C++ program to
   a. add/ subtract two complex numbers and
   b. add/ subtract a real number to a complex number.
4. Using template function, Perform a generic sorting program and demonstrate the same for integers and real numbers.
5. Overloading the operators ++ and --, Implement a C++ program to create a stack of integers and demonstrate the push and pop operations.
6. Overloading the operators + and -, write a C++ program to
   a. Subtract two given dates and find the difference in days.
   b. Add given number of days to a given date
7. Using friend function concept, demonstrate a C++ program

8. Create a class called A and derive two classes B and C from this. Demonstrate single inheritance with suitable functions.
9. With the help of a Virtual function, demonstrate a C++ program that displays the area and circumference of a Rectangle, Square and a Circle.
10. Implement a C++ program to find the area of rectangle, triangle and sphere. Use function overloading concept
11. Design a C++ program to create aclass STUDENT with the following specification: Data members: Name, Roll_no, SGPA. Member Functions: Read and Write. Use these specification to read and print the information of 3 students.
12. Implement a C++ program to
    a. Concatenate two strings, and
    b. Search if the second string is present in the first string.

Text Books:

1. Object- oriented programming with C++,E Balguruswamy, 6th Edition, Tata McGraw Hill, 2014
2. Mastering C++, K R Venugopal, Rajkumar Buyya, Tata McGraw Hill, 2nd Edition, Tata McGraw Hill, 2013.

Reference Books:

1. The Complete Reference C++, Herbert Scheldt, 4th Edition, Tata McGraw Hill, 2012
2. C++ Primer, Stanley B.Lippman, JoseeLajoie, 5th Edition, Pearson Education, 2009.

MOOC Course:

3. https://onlinecourses.nptel.ac.in/noc19_cs38/preview

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3													
CO2		2											3	
CO3		3											3	

Course Title	**DISCRETE MATHEMATICAL STRUCTURES AND COMBINATORICS**		
Course Code	**22IS306C**	**L-T-P**	**(2-2-0)3**
Exam	**3 Hrs.**	**Hours/Week**	**4**
SEE	**50 Marks**	**Total Hours**	**40**

Course Objective: Introduction of Discrete structures and principle of Combinatorics which may be employed as tools in the applications of Computer Science & Information Technology.

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

#	Course outcomes	Mapping to PO's	Mapping to PSO's
1	Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers	1,2	-
2	Apply induction hypotheses, various methods of proof methods in decision taking problems.	1,2	-
3	Solve combinatorial problems using counting principles.	1,2	-
4	Apply graph concepts to model and analyse problems of information science & engineering	1,2	-

MODULE-1 — 10 Hrs.

Set theory: A review of set operations, statements of laws of set theory, verification of set identities using Venn diagrams and Membership table. Illustrated examples.

Counting principles: Rule of addition, multiplication principle. Inclusion and Exclusion Principle for 2 and 3 sets, principles of permutation, combination, generalized permutation and generalized combination principle. Illustrative examples.

MODULE-2 — 10 Hrs.

Fundamentals of logic: Basic logic connectives and truth tables. Logical equivalence and Tautologies. Statement of laws o logic. Logic implication - Rules of inference theory.

Methods of proof: Using rules of inference theory, methods of direct and indirect proof.

MODULE-3 — 10 Hrs.

Relations: Cartesian products and relations, computer representation of a relation and directed graph, properties of relations, equivalence relations and partitions. Partially ordered set and Hasse diagram.

Functions: Definition, various types of functions - one to one function, onto function, bijective function, invertible functions and function composition Application of Stirling numbers of second kind.

MODULE-4 — 10 Hrs.

Graph theory: Basic terminologies of a graph. discussion of connected and disconnected graphs, Euler and Hamilton graphs, Planar graphs and Graph colouring.

Trees: Definition, properties of a tree. weighed trees, prefix codes and biconnected components. Modelling of real-life problems using graphical approach and their analysis.

Text Books:

1. Discrete and Combinatorial Mathematics, R C Grimaldi, Pearson's publications, 5th edition, 2007.

Reference Books:

1. Discrete Mathematical Structures, by D. S. Malik & M. K. Sen, Thomson's Publications, First edition, 2006.

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]												
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2												
CO2	3	2												
CO3	3	2												

CO4	**3**	**2**												

Course Title	**GRAPH THEORY AND COMBINATORICS**		
Course Code	**22IS306D**	**L-T-P**	**(2-2-0)3**
Exam	**3 Hrs.**	**Hours/Week**	**4**
SEE	**50 Marks**	**Total Hours**	**40**

Course Objective: Students will be able to understand combinatorics, graphs, trees and their applications.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Identify and explain the properties of various types of graphs	1, 2	
2	Construct different types of trees, identify biconnected components and articulation points	1, 2	
3	Apply the generalized principle of Inclusion and Exclusion theorem	1, 2	
4	Apply the concept of generating functions to solve the given problems of counting theory	1, 2	

MODULE – 1

10 Hrs.

Introduction to Graph Theory: Definitions and Examples, Sub-graphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Circuits and Trails.

Planar Graphs & Graph Coloring: Definition and Examples, A Discussion on Kuratowski Graphs, Detection of Planarity of a Graph.

MODULE – 2

10 Hrs.

Dual Graphs and Properties of a Dual Graph with Respect to a Planar Graph: Chromatic Number and Chromatic Polynomial of a Graph.

Trees: Definition, Properties, and Examples. Rooted Trees and Binary Trees, Weighed Trees and Prefix Codes. Optimization and Matching: Transport Networks - Max-Flow Min-cut Theorem.

MODULE -3

10 Hrs.

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

The Catalan Numbers. Pigeon Hole Principle, Advanced Counting Techniques: The Principle of Inclusion and Exclusion – Definition and Illustrative Examples, Generalizations of the Principle

MODULE -4

10 Hrs.

Derangements – Nothing Is In Its Right Place, Rook Polynomials, Arrangements with Forbidden Positions, Generating functions, introductory examples, Definition and examples.

Calculational Techniques, Partitions of Integers, the Exponential Generating Function, the Summation Operator.

Text Books:

1. R C Grimaldi, Discrete Mathematics and Combinatorics, Pearson Education, 5th Edition, 2007.

Reference Books:

1. Narsing Deo, Graph Theory with Applications to Engineering and Computer Science, PHI Publications

5. V. Balakrishnan, Combinatorics, Schaum Series, Tata McGraw-Hill Publications

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2												
CO2	3	2												

CO3	**3**	**2**												
CO4	**3**	**2**												

Course Title	**R PROGRAMMING**		
Course Code	**22IS307A**	**L-T-P**	**(1-0-0)1**
Exam	**3 Hrs.**	**Hours/Week**	**2 Hrs**
SEE	**50 Marks**	**Total Hours**	**28**

Course Objective: Students will be able to learn and practice programming techniques using R programming.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Explain the fundamental syntax of R data types, expressions and the usage of the R-Studio IDE	1	-
2.	Develop a program in R with programming constructs: conditionals, looping and functions.	3	-
3.	Apply the list and data frame structure of the R programming language.	2	-
4.	Use visualization packages and file handlers for data analysis.	3	-

S.N	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. Assign different types such as Double, Integer, Logical, Complex and Character and understand the difference between each data type. b) Demonstrate Arithmetic and Logical Operations with simple examples. c) Demonstrate generation of sequences and creation of vectors. d) Demonstrate Creation of Matrices e) Demonstrate the Creation of Matrices from Vectors using Binding Function. f) Demonstrate element extraction from vectors, matrices and arrays **Suggested Reading** – Text Book 1 – Chapter 1 (What is R, Installing R, Choosing an IDE – RStudio, How to Get Help in R, Installing Extra Related Software), Chapter 2 (Mathematical Operations and Vectors, Assigning Variables, Special Numbers, Logical Vectors), Chapter 3 (Classes, Different Types of Numbers, Other Common Classes, Checking and Changing Classes, Examining Variables)
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. **Suggested Reading** – Text Book 1 – Chapter 4 (Vectors, Combining Matrices)
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

Suggested Reading – Text Book 1 – Chapter 4 (Matrices and Arrays – Array Arithmetic)

4 Develop a program to find the factorial of given number using recursive function calls.

Suggested Reading – Reference Book 1 – Chapter 5 (5.5 – Recursive Programming) Text Book 1 – Chapter 8 (Flow Control and Loops – If and Else, Vectorized If, while loops, for loops), Chapter 6 (Creating and Calling Functions, Passing Functions to and from other functions)

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.

Suggested Reading – Reference Book 1 - Chapter 5 (5.5 – Recursive Programming)

Text Book 1 – Chapter 8 (Flow Control and Loops – If and Else, Vectorized If, while loops, for loops), Chapter 6 (Creating and Calling Functions, Passing Functions to and from other functions)

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.

Suggested Reading – Text Book 1 –Chapter 12 – (Built-in Datasets) Chapter 14 – (Scatterplots) Reference Book 2 – 13.2.5 (Covariance and Correlation)

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
1005	CD Drive	800

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

Suggested Reading –Reference Book 2 – Chapter 7 (7.4 – The ggplot2 Package), Chapter 24 (Smoothing and Shading)

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"

Suggested Reading – Text Book 1 – Chapter 12(CSV and Tab Delimited Files)

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 from 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 carburettors.

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? **References (Web links):** 1. https://cran.r-project.org/web/packages/explore/vignettes/explore_mtcars.html 2. https://www.w3schools.com/r/r_stat_data_set.asp 3. https://rpubs.com/BillB/217355
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. **Suggested Reading** – Reference Book 2 – Chapter 20 (General Concepts, Statistical Inference, Prediction)

Text Books:

1. Cotton, R., Learning R: a step by step function guide to data analysis. 1st edition. O'reilly Media Inc

Reference Books:

1. Jones, O., Maillardet. R. and Robinson, A. (2014). Introduction to Scientific Programming and Simulation Using R. Chapman & Hall/CRC, The R Series.
2. Davies, T.M. (2016) The Book of R: A First Course in Programming and Statistics. No Starch Press.

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]												
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	**3**													
CO2			**2**											
CO3		**2**												
CO4			**2**											

Course Title	**Data Analytics with Excel**		
Course Code	**22IS307B**	**L-T-P**	**(0-0-1) 1**
Exam	**3 Hrs.**	**Hours/Week**	**2 Hrs**
SEE	**50 Marks**	**Total Hours**	**28**

Course Objective: Students will be able to identify the principles of data analysis and apply analysis techniques to datasets in Excel.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Understand the key concepts of data analysis, its significance in various domains, and the role of Excel as a tool for data manipulation and visualization.	1	1
2.	Demonstrate the ability to apply Excel functions and tools for data manipulation, cleaning, and transformation to analyse and gain insights from datasets.	2,3,5,9	1
3.	Develop the skills to analyse datasets using descriptive statistics, data visualization techniques, and Excel charts to effectively communicate patterns and trends.	2,4,5	1

1. Introduction to Data Analysis and Excel Basics

Create a simple Excel spreadsheet that tracks personal expenses over the course of a week. Use basic functions to calculate total expenses.

2. Data Entry, Cleaning, and Data Types

1. Given a dataset with missing values, fill in the missing values using appropriate Excel techniques.
2. Create a dataset with a mix of data types (numeric, text, dates) and ensure data consistency.

3. Descriptive Statistics and Data Visualization

1. Analyse a sales dataset by calculating mean, median, mode, and range using Excel functions.
2. Create a bar chart to visualize the distribution of sales across different products.

4. Data Analysis Functions and Pivot Tables

1. Analyse a student performance dataset to calculate the average score, highest score, and lowest score using Excel functions.
2. Create a pivot table to summarize and compare the sales figures for different regions.

5. Exploratory Data Analysis

1. Analyse a dataset containing employee information. Use filtering and sorting to find the top earners and junior employees.
2. Apply data validation to ensure that dates are entered correctly in a dataset.

6. Hypothesis Testing and Regression Analysis

1. Compare the average test scores of two groups of students using a t-test in Excel.
2. Perform a simple linear regression analysis to predict sales based on advertising spending.

7. Advanced Data Visualization

Create a scatter plot to show the relationship between hours studied and exam scores using a dataset of your choice.

8. Advanced Functions for Data Analysis

Analyse a real-world dataset using advanced Excel functions such as VLOOKUP and COUNTIFS.

9. Working with Large Datasets

Analyse a large dataset (provided) using Excel's filtering, sorting, and pivot table features.

10. Time Series Analysis

Analyse a time series dataset (provided) using Excel to identify trends and patterns.

11. Advanced Chart Types

Create a histogram to show the distribution of ages in a dataset using Excel.

12. Formula Auditing and Error Handling

Identify and rectify errors in each dataset using Excel's formula auditing tools.

Text Books:

3. "Excel Data Analysis for Dummies" by Stephen L. Nelson and E. C. Nelson: A beginner-friendly guide to using Excel for data analysis.
4. "Data Analysis Using SQL and Excel" by Gordon S. Linoff: Covers data analysis techniques using Excel and SQL.

Reference Books:

1. "Data Smart: Using Data Science to Transform Information into Insight" by John W. Foreman

MOOC Course:

https://elearn.nptel.ac.in/shop/nptel/digital-skilling/

Course Articulation Matrix

Course Outcomes	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3												3	
CO2		3	2		2				2				3	
CO3		3		2	2								3	

Course Title	Data Visualization with Python		
Course Code	22IS307C	L-T-P	(0-0-1)1
Exam	3 Hrs.	Hours/Week	2 Hrs
SEE	50 Marks	Total Hours	28

Course Objective: Students will be able to demonstrate the use of IDLE or PyCharm IDE to create Python Applications
Course outcomes: At the end of course, student will be able to:

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Demonstrate the use of IDLE or PyCharm IDE to create Python Applications.	2	-
2.	Use Python programming constructs to develop programs for solving real-world problems.	2	2
3.	Use Matplotlib for drawing different Plots.	5	-
4.	Demonstrate working with Seaborn, Bokeh for visualization.	2	-
5.	Use Plotly for drawing Time Series and Maps.	5	-

SN	Experiments
1	a) Write a python program to find the best of two test average marks out of three test’s marks accepted from the user. b) Develop a Python program to check whether a given number is palindrome or not and also count the number of occurrences of each digit in the input number. Datatypes: https://www.youtube.com/watch?v=gCCVsvgR2KU Operators: https://www.youtube.com/watch?v=v5MR5JnKcZI Flow Control: https://www.youtube.com/watch?v=PqFKRqpHrjwFor loop: https://www.youtube.com/watch?v=0ZvaDa8eT5s While loop: https://www.youtube.com/watch?v=HZARImviDxg Exceptions: https://www.youtube.com/watch?v=6SPDvPK38tw
2	a) Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed. b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions. Functions:https://www.youtube.com/watch?v=BVfCWuca9nw Arguments:https://www.youtube.com/watch?v=ijXMGpoMkhQ Return value: https://www.youtube.com/watch?v=nuNXiEDnM44
3	a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters. b) Write a Python program to find the string similarity between two given strings Sample Output: Original string: Python Exercises Python Exercises Similarity between two said strings: Sample Output: Original string: Python Exercises Python Exercise Similarity between two said strings:1.0 0.967741935483871 Strings: https://www.youtube.com/watch?v=lSItwlnF0eU String functions: https://www.youtube.com/watch?v=9a3CxJyTq00
4	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.

	https://www.youtube.com/watch?v=RRHQ6Fs1b8w&list=PLjVLYmrlmjGcC0B_FP3bkJJIPkV5GuZR&index=3 https://www.youtube.com/watch?v=7ABCuhWO9II&list=PLjVLYmrlmjGcC0B_FP3bkJJIPkV5GuZR&index=4
5	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. https://www.youtube.com/watch?v=Qk7caotaQUQ&list=PLjVLYmrlmjGcC0B_FP3bkJJIPkV5GuZR&index=6 https://www.youtube.com/watch?v=PSji21jUNO0&list=PLjVLYmrlmjGcC0B_FP3bkJJIPkV5GuZR&index=7
6	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. https://www.youtube.com/watch?v=UO98lJQ3QGI&list=PL-osiE80TeTvipOqomVEeZ1HRrcEvtZB_
7	Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions. https://www.youtube.com/watch?v=6GUZXDef2U0
8	Write a Python program to explain working with bokeh line graph using Annotations and Legends. a) Write a Python program for plotting different types of plots using Bokeh. https://www.youtube.com/watch?v=HDvxYoRadcA
9	Write a Python program to draw 3D Plots using Plotly Libraries. https://www.youtube.com/watch?v=cCck7hCanpw&list=PLE50-dh6JzC4onXqkv9H3HtPbBVA8M94&index=4
10	a) Write a Python program to draw Time Series using Plotly Libraries. b) Write a Python program for creating Maps using Plotly Libraries. https://www.youtube.com/watch?v=xnJ2TNrGYik&list=PLE50-dh6JzC4onXqkv9H3HtPbBVA8M94&index=5 https://www.youtube.com/watch?v=D35m2CdMhVs&list=PLE50-dh6JzC4onXqkv9H3HtPbBVA8M94&index=6

Text Books:

1. Al Sweigart, “Automate the Boring Stuff with Python”,1stEdition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
2. Reema Thareja “Python Programming Using Problem Solving Approach” Oxford University Press.
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”, 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf)
4. Jake VanderPlas “Python Data Science Handbook” 1st Edition, O’REILLY.

Course Articulation Matrix

Course OutcomeS	Program Outcomes [POs]												
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1		**2**												
CO2		**2**												2
CO3					**3**									
CO4		**2**												

CO5					3									

Course Title	**PROJECT MANAGEMENT WITH GIT**		
Course Code	**22IS307D**	**L-T-P**	**(0-0-1)1**
Exam	**3 Hrs.**	**Hours/Week**	**2 Hrs**
SEE	**50 Marks**	**Total Hours**	**28**

Course Objective: Students will be familiarized with basic commands of Git and understand how to collaborate and work with Remote Repositories.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Use the basics commands related to git repository	1	-
2.	Create and manage the branches	1	-
3.	Apply commands related to Collaboration and Remote Repositories	2	-
4.	Use the commands related to Git Tags, Releases and advanced git operations	1	-
5.	Analyse and change the git history	2	

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?
1	**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."

1	**Analysing and Changing Git History** Write the command to display the last five commits in the repository's history.
1	**Analysing and Changing Git History** Write the command to undo the changes introduced by the commit with the ID "abc123".

Text Books:

1. Prem Kumar Ponuthorai, Jon Loeliger, Version Control with GiT, 3rd edition, O'Reilly Media, Inc.

Course Articulation Matrix

Course Outcomes	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	**3**													
CO2	**3**													
CO3		**2**												
CO4	**3**													
CO5		**2**												

Course Title	**SOCIAL CONNECT & RESPONSIBILITY**		
Course Code	**22SCR**	**L-T-P**	**(0-0-2)1**
Exam	**3 Hrs.**	**Hours/Week**	**2**
SEE	**50 Marks**	**Total Hours**	**15**

Course Objective: Provide a formal platform for students to communicate and connect with their surroundings and create a responsible connection with society

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

#	**Course Outcomes**	**Mapping to PO's**	**Mapping to PSO's**
1	Describe societal challenges and build solutions to alleviate these complex social problems through immersion, design & technology.	6	-
2	Communicate and connect with their surroundings.	7, 12	-

MODULE – 1

Plantation and adoption of a tree: Plantation of a tree that will be adopted by a group of students. They will also make an excerpt either as a documentary or a photo blog describing the plant’s origin, its usage in daily life, and its appearance in folklore and literature.

MODULE – 2

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, photoblog and documentary on evolution and practice of various craft forms.

MODULE -3

Organic farming and waste management: Usefulness of organic farming, wet waste management in neighboring villages, and implementation in the campus.

MODULE -4

Water Conservation: knowing the present practices in the surrounding villages and implementation in the campus, documentary or photo blog presenting the current practices. Food Walk City’s culinary practices, food lore, and indigenous materials of the region used in cooking.

Course Conduction

A total of 14-20 hrs engagement per semester is required for the course. Students will be divided into teams and each team will be handled by two**faculty mentors**. Faculty mentors will design the activities for evaluation.

Guideline for Assessment Process:

Continuous Internal Evaluation (CIE)

After completion of, the social connect, the student shall prepare, with daily **diary** as reference, a comprehensive report in consultation with the mentor/s to indicate what he has observed and learned in the social connect period. The report should be signed by the mentor. The report shall be evaluatedon the basis of the following criteria and/or other relevant criteria pertaining to the activity completed.

Marks allotted for the diary are out of 50. Planning and scheduling the social connect Information/ Data collected during the social connectAnalysis of the information/data and report writing

Considering all above points allotting the marks as mentioned below

Excellent	80 to 100

Good	60 to 79
Satisfactory	40 to 59
Unsatisfactory and fail	<39

Course Title	**SOFTWARE ENGINEERING**		
Course Code	**22IS401**	**(L-T-P)C**	**(2-2-0) 3**
Exam	**3 Hrs.**	**Hours/Week**	**4**
SEE	**50 Marks**	**Total Hours**	**40**

Course Objective:

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Analyze a system for identifying the software requirements	2	2
2	Apply software process activities during system design	3	2
3	Apply different approaches of verifying and validating a software product.	1	2
4	Apply various project management activities	3	2

MODULE – 1	10 Hrs.

Introduction: Professional Software Development, Software Engineering Ethics. Case studies, **Software Processes:** Software Processes models, Process activities, coping with change, The Rational Unified Process. **Agile Software Development:** Agile Methods, Plan-driven and Agile Development, Extreme programming. Agile Project Management. Scaling Agile Methods.

Self Study: Case Studies- A patient information system for mental health care, A wilderness weather station

MODULE – 2	10 Hrs.

Requirements Engineering: Functional and non-functional requirements. The software Requirements Document. Requirements Specification, Requirements Engineering Processes. Requirements Elicitation and Analysis. Requirements validation. Requirements Management. **System Models:** Context models. Interaction models. Structural models. Behavioural models. Model-driven engineering. **Self Study:** Behavioural models

MODULE -3	10 Hrs.

Design and Implementation: Object-oriented design using the UML, Design patterns, Implementation issues, Open source development. **Software Testing:** Development testing, Test-driven development, Release testing, User testing. **Software Evolution**: Evolution processes, Program evolution dynamics. Software maintenance, Legacy system management.

Self Study: Designing UML diagrams

MODULE -4	10 Hrs.

Software Project Management: Software Project Management Complexities, Responsibilities of a software project Manager, Project Planning, Metrics for project size estimation, Project estimation techniques, Empirical estimation techniques, Scheduling, Organization and Team Structures, Staffing, Risk Management.

Self Study: COCOMO, ISO9000, SEI Capability Maturity Model, Other Important Quality Standards, Six Sigma.

Text Books:

1. Ian Sommerville, "Software Engineering", 9th Edition, Person Education, 2014. (Chapters:1,2,3,4,5,7,8,9)
2. Fundamentals of Software Engineering, Rajib Mall, 2015, Prentice-Hall Of India Pvt. Ltd., ISBN: 9788120348981 (Chapters: 3)

Reference Books:

1. Roger S. Pressman, "Software Engineering - A Practitioners Approach", 7th Edition, McGraw-Hill, 2007.
2. Waman S. Jawadekar, "Software Engineering Principles and Practice", Tata McGraw-Hill, 2004.
3. Software Engineering: A Concise introduction to Software Engineering by Pankaj Jalot, Springer.

Tutorial Component

Students will be made to understand Case studies on topics discussed in classes.Presentation on case studies. This will help students in better understanding of concepts.

1. Considering the following case studies,

a) Identify the requirements and prepare the SRS document (as per IEEE format) from Problem Statements.

b) Design Models using following UML diagrams for the case studies given below (Tool: Star UML/Enterprise Architect)

c) Use of any Open Source Test Tool like Selenium or equivalent as determined by the course co-coordinator

Structural Diagrams

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

Behavioural Diagrams

- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram

List of Case Studies

1. Library Management System 2. Hospital Management System 3. Online reservation Management System 4. Airport check-in and security screening System 5. Restaurant business System 6. Bank ATM System 7. Ticket vending machine 8. Student marks Analyzing System

MOOC Course:

4. Software Engineering https://nptel.ac.in/courses/106/105/106105182/

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1		2												2
CO2			3											2
CO3	2													2
CO4			2											2

Course Title	**DATABASE MANAGEMENT SYSTEMS**		
Course Code	**22IS403**	**(L-T-P)C**	**(3-0-2) 4**
Exam	**3 Hrs.**	**Hours/Week**	**5**
SEE	**50 Marks**	**Total Hours**	**50 (36L+14P)**

Course Objective: Students will acquire the concepts of databases, and application of SQL for solving problems.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Understand the concepts of Database Management Systems and its applications.	1	-
2	Design ER diagram for real world applications and develop SQL queries.	3	2
3	Apply normalizations for relation scheme.	2	-
4	Describe the issues in transaction management.	1	-

Module-1 **9 Hrs**

Introduction: Introduction, an example, Characteristics of Database approach, Actors on the Screen, Workers Behind the Scene, Advantages of Using DBMS Approach. Data Models, Schemas and Instances, Three-schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment.

Entity-Relationship Model: Using High-Level Conceptual Data Models for Database Design, An Example Database Application, Entity Types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design, ER Diagrams, Naming Conventions and Design Issues, Relationship Types of Degree Higher Than Two.

Module-2 **9 Hrs**

Relational Model and Relational Algebra: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, Transactions and Dealing with Constraint Violations, Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relational Operations: JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra, Relational Database Design Using ER- to-Relational Mapping.

SQL: SQL Data Definition and Data Types, Specifying Basic Constraints in SQL, Basic Queries in SQL.

Module-3 **9 Hrs**

SQL(contd.): More Complex SQL Queries, Insert Delete and Update Statements in SQL, Specifying Constraints as Assertion and Trigger, Views (Virtual Tables) in SQL, Schema Change Statements in SQL.

Database Design: Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Module-4 **9 Hrs**

Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock - Based Concurrency Control, Performance of Locking, Transaction Support in SQL, Introduction to Crash Recovery. Crash Recovery: Introduction to ARIES

NoSQL: An overview of NoSQL, Characteristics of NoSQL, NoSQL storage types, Advantages and Drawbacks of NoSQL,.

Practical Component:

Perform the following programs using MySQL.

1. Consider the following schema:

EMPLOYEE (Ename, Ssn, Bdate, Sex, Address, salary, Mgrssn, Dno)

DEPARTMENT (Dname, Dnumber, Mgrssn, Mgr_start_date)

PROJECT (Pname, Pnumber, Plocation, Dnum)

WORKS_ON (Essn, Pno, Hours)

DEPENDENT(Essn, Dependent_name,Sex)

Create above tables by specifying primary key, foreign key and other suitable constraints.
Insert atleast 5 tuples to each created table.
i. Retrieve the name and address of all employees who work for the "ISE" department.
ii. For each employee, retrieve the employee's name and the name of his or her immediate supervisor
iii. Find the sum of all salaries of all employees
iv. For each department, retrieve the department number, the number of employees in the department and their average salary.

2. Consider the following relation schema:
SAILORS (Sid: integer, Sname: string, Rating: integer, Age: real)
BOATS (Bid: integer, Bname: string, Color: string)
RESERVES (sid: integer, Bid: integer, Day:date)
Create above tables by specifying primary key, foreign key and other suitable constraints.
Insert atleast 5 tuples to each created table.
Design a database to the satisfy the above requirements and answer following queries
i. Find all sailors with a rating above 7
ii. Find the names of sailors who have reserved boat number 103
iii. Find the names of sailors who have reserved a red boat
iv. Find the names of sailors who have reserved a red or a green boat
3. Consider the following relation schema:
STUDENT (Snum: integer, Sname: string, Major: string, Level: string, Age: integer)
CLASS (Cname: string, Meets at: string, Room: string, Fid: integer)
ENROLLED (Snum: integer, Cname: string)
FACULTY (Fid: integer, Fname: string, Deptid: integer)
The meaning of these relations is straightforward; for example, enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc)

3. Write the following queries in SQL. No duplicates should be printed in any of the answers.
i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith
ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
iii. Find the names of all students who are enrolled in two classes that meet at the same time.
iv. Find the names of faculty members who teach in every room in which some class is taught.

4. Consider the relation schema for book dealer database:
AUTHOR (Author-id:int, Name:string, City:string, Country:string)
PUBLISHER (Publisher-id:int, Name:string, City:string, Country:string)
CATALOG (Book-id:int, Title:string, Author-id:int, Publisher-id:int, Category-id:int, Year:int, Price:int)
CATEGORY (Category-id:int, Description:string)
ORDER-DETAILS (Order-no:int, Book-id:int, Quantity:int)
Create the above tables by properly specifying the primary keys and the foreign keys. Enter atleast five tuples for each relation.
i. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
ii. Find the author of the book which has maximum sales.
iii. Demonstrate how you increase the price of books published by a specific publisher by 10%
iv. List any department that has all its adopted books published by a specific publisher

5. Consider the schema for Movie Database:
ACTOR (Act_id, Act_Name,Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)
MOVIE_CAST (Act_id, Mov_id, Role)
RATING (Mov_id, Rev_Stars)
Write SQL queries to
Create the above tables by properly specifying the primary keys and the foreign keys. Enter atleast five tuples for each relation.

1. List the titles of all movies directed by ‘Hitchcock’.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015.
4. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

6. Consider the following database for a banking enterprise
BRANCH (branch-name: String, branch-city: String, assets: real)
ACCOUNTS (accno: int, branch-name: String, balance: real)
DEPOSITOR (customer-name: String, customer-street: String, customer-city: String)
LOAN (loan-number: int, branch-name: String, amount: real)
BORROWER (customer-name: String, loan-number: int)
Create the above tables by properly specifying the primary keys and the foreign keys. Enter atleast five tuples for each relation.

1. Find all the customers who have at least two accounts at the Main branch.
2. Find all the customers who have an account at all the branches located in a specific city.
Demonstrate how you delete all account tuples at every branch located in a specific city.

Text Books:

1. Elmasri and Navathe,: Fundamentals of Database Systems", 7th Edition, Addison-Wesley,2015.
2. Raghu Ramakrishnan and Johannes Gehrke," Database Management Systems", 3rd Edition, McGraw-Hill,2007

Reference Books:

1. Silberschatz, Korth and Sudarshan: “Database System Concepts”, 6th Edition, Mc-Graw Hill, 2010.
2. C.J. Date, A. Kannan, S. Swamynatham: “An Introduction to Database Systems”, 8th Edition, Pearson Education, 2006.

MOOC:

1. Database Management Systemshttps://nptel.ac.in/courses/106/105/106105175/

Course Articulation Matrix

Course Outcomes	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2			3											2
CO3		3												
CO4	3													

Course Title	DESIGN AND ANALYSIS OF ALGORITHMS		
Course Code	22IS404	(L-T-P)C	(3-0-2)4
Exam	3hrs	Hours/Week	5
SEE	50 Marks	Total Hours	50 (36L+14P)

Course Objective: Students will be able to design algorithms using various strategies and analyze it mathematically.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Apply various algorithm design techniques to solve the given problem.	3	1
2.	Analyse the time complexity of the algorithm using asymptotic notations.	2	1
3.	Differentiate tractable & intractable problems & apply techniques that help to cope up with limitation of algorithm power.	3	1
4.	Conduct experiments to implement the designed algorithms	3	1

MODULE – 1 9 Hrs.

Introduction: Notion of Algorithm, Fundamentals of algorithmic problem solving. **Fundamentals of the Analysis of Algorithm Efficiency:** Analysis framework, Asymptotic notations and Basic efficiency classes, Mathematical analysis of Recursive and Non-recursive algorithms, Examples. **Brute Force:** Selection Sort and Bubble Sort, Sequential Search and String Matching, Exhaustive search.

MODULE – 2 9 Hrs.

Divide-and-Conquer: Binary Search, Merge Sort, Quick Sort, Binary tree traversals and related properties, Multiplication of large integers, Strassen's Matrix multiplication. **Decrease-and-Conquer:** Insertion Sort, Depth First and Breadth First Search, Topological sorting, Algorithms for generating combinatorial objects.

MODULE -3 9 Hrs.

Transform-and-Conquer: Pre-sorting, Balanced Search Trees, Heaps and Heap Sort, Problem reduction. **Space and Time Trade-off**: Sorting by counting, Input enhancement in string Matching (only Horspool), Hashing. **Dynamic Programming**: Computing a Binomial coefficient, Warshall's Algorithm, Floyd's algorithms, The Knapsack problem.

MODULE -4 9 Hrs.

Greedy Technique: Prim's algorithm, Kruskal's algorithm, Dijkstra's algorithm, Huffman trees, **Limitations of Algorithm Power:** Lower-bound arguments, Decision trees, P, NP and NP-Complete Problems, coping with the Limitations of Algorithm Power: Backtracking, Branch-and-bound.

Practical Component/Tutorial:

Guided Experiments

Implement the following using C/Java Language.

1. Employees in an organization need to be grouped for a tournament based on their ages. Sort the ages using Merge sort and find the time required to perform the sorting.
2. Students in a department need to be selected for a high jump competition based on their height (integer values only). Sort the heights of students using Quick sort and find the time required for the sorting.
3. Print all the nodes reachable from a given starting node in a graph using Depth First Search method and Breadth First Search. Also check whether a graph is connected.
4. Obtain the topological ordering of vertices in a given digraph.
5. Implement Horspool algorithm for String Matching.
6. Sort a given set of elements using the Heap sort method.
7. Implement Floyd's algorithm and Warshall's algorithm for a given graph.

8. There are n different routes from hostel to college. Each route incurs some cost. Find the minimum cost route to reach the college from the hostel using Prim’s algorithm.
9. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal’s algorithm
10. Implement 0/1 Knapsack problem using dynamic programming.
11. Implement N Queen’s problem using Backtracking.

Open ended Experiments

Students have to solve a given problem using any suitable design technique and demonstrate its efficiency. Sample list of problems (but not limited to this) that can be considered are

1. Josephus problem

2.Travelling salesman problem

3. Job assignment problem

4.Boyre Moore string matching algorithm

5. Searching problem like - Given a string, find all possible palindromic substrings in it, Given a sequence of numbers between 2 and 9, print all possible combinations of words formed from the mobile keypad which has English alphabets associated with each key.

6. Sorting problem like - Given two integer arrays, reorder elements of the first array by the order of elements defined by the second array.

Text Books:

1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition, Pearson Education, 2017.

Reference Books:

1. Thomas H. Coremen, Charles E. Leiserson, Ronald L. Rivest, Introduction to Algorithms, 3rd Edition PHI
2. Horowitz E., Sahani S., Rajasekharan S., Computer Algorithms, Galgotia Publications

MOOC Course:

1. Design and Analysis of Algorithms https://nptel.ac.in/courses/106/106/106106131/

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]												
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1			3										3	
CO2		3											3	
CO3			3										3	
CO4			3										3	

Course Title	**C# AND .NET TECHNOLOGIES**		
Course Code	**22IS405A**	**(L-T-P)C**	**(2-0-2) 3**
Exam	**3 Hrs**	**Hours/Week**	**4**
SEE	**50 Marks**	**Total Hours**	**26L + 14P**

Course Objective: Students will be able to apply Object Oriented Programming concepts for designing Applications using language C# and IDE – Visual Studio.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Develop C# programs using Visual Studio IDE.	1,2	-
2	Apply Object Oriented Programming concepts in C# programming language	1,2	-
3	Interpret Interfaces and define custom interfaces for application.	1,2	-
4	Analyse a C# program for identifying bugs.	2	-

MODULE – 1	7Hrs.

Introducing Microsoft Visual C# and Microsoft Visual Studio 2015: Welcome to C#, working with variables, operators, and expressions, writing methods, and applying scope, using decision statements, using compound assignment and iteration statements, Managing errors and exceptions.

MODULE – 2	7Hrs.

Understanding the C# object model: Creating and Managing classes and objects, understanding values and references, creating value types with enumerations and structures, Using arrays.

MODULE -3	6Hrs.

Understanding parameter arrays, working with inheritance, creating interfaces and defining abstract classes, Using garbage collection and resource management.

MODULE -4	6Hrs.

Defining Extensible Types with C#: Implementing properties to access fields, introducing generics, Using collections, Operator overloading.

Practical Component/Tutorial:

1. Write a C# program that calculates the area of a rectangle. The program should ask the user to enter the length and width of the rectangle as inputs. Then, using appropriate variables, operators, and expressions, calculate and display the area of the rectangle.
2. Program that converts a temperature from Celsius to Fahrenheit. The program should have a method called Convert To Celsius that takes a temperature in Celsius as a parameter and returns the corresponding temperature in Fahrenheit. Use appropriate variable types and apply appropriate scope in your program.
3. Write a C# program that calculates the average of a series of numbers entered by the user. The program should ask the user to enter the numbers one by one, and use a loop to accumulate the sum of the numbers. If the user enters a non-numeric value, the program should handle the exception gracefully and continue asking for valid input. Once the user is done entering numbers, calculate and display the average.
4. Create a class called "Car" that represents a car object. The Car class should have properties such as "Make", "Model", and "Year" to store the car's make, model, and manufacturing year respectively. Implement a method called "StartEngine" that prints a message indicating that the car's engine has started. In the Main method, create an instance of the Car class, set its properties, and call the StartEngine method to demonstrate the usage of classes and objects.
5. Create an enumeration called "DaysOfWeek" that represents the days of the week (e.g., Monday, Tuesday, etc.). Then, create a structure called "Appointment" that has properties such as "MeetingName" (string), "Day" (DaysOfWeek), and "Time" (DateTime). In the Main method, create an array of Appointment objects and populate it with different appointments. Display the details of each appointment using a loop.
6. Write a program that takes a variable number of integers as input using a parameter array. The program should calculate and display the sum of all the integers entered by the user.
7. Create a base class called "Shape" that has an abstract method called "CalculateArea" and a virtual method called "DisplayInfo". Create two derived classes called "Rectangle" and "Circle" that inherit from the Shape class. Implement the CalculateArea method in both derived classes to calculate the area of a rectangle and a circle respectively.

Implement the DisplayInfo method in each derived class to display information about the shape. Finally, create instances of the Rectangle and Circle classes, call the CalculateArea and DisplayInfo methods on them to demonstrate inheritance and polymorphism.

8. Create a class called "Person" that has private fields for name and age. Implement properties to access and modify these fields. The Name property should be read-only, while the Age property should be read-write. In the Main method, create an instance of the Person class, set the name and age properties, and display the person's details.
9. Create a generic class called "Stack<T>" that represents a stack data structure. The class should have methods to push an item onto the stack, pop an item from the stack, and check if the stack is empty. In the Main method, create two instances of the Stack class, one for storing integers and another for storing strings. Push some items onto both stacks and perform pop operations to demonstrate the generic behaviour of the class.
10. Create a class called "Complex" that represents a complex number. Implement operator overloading for addition, subtraction, and multiplication of complex numbers. Test the operator overloading by performing arithmetic operations on instances of the Complex class.

Text Books:

1. John Sharp, Microsoft Visual C# Step by Step, 8th Edition, PHI Learning Pvt. Ltd. 2016

Reference Books:

1. Tom Archer, Andrew Whitechapel, Inside C#, WP Publishers
2. Herbert Schildt, The Complete Reference C# 3.0, Tata McGraw Hill Education Private Limited

MOOC Course:

1. https://www.coursera.org/learn/intro-to-dotnet-core

Course Articulation Matrix

Course Outcomes	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	**3**	**3**												
CO2	**2**	**2**												
CO3	**2**	**2**												
CO4		**2**												

Course Title	**INTERNET OF THINGS**		
Course Code	**22IS405B**	**(L-T-P)C**	**(3-0-0)3**
Exam	**3 Hrs.**	**Hours/Week**	**3**
SEE	**50 Marks**	**Total Hours**	**40 Hrs**

Course Objective: Students will be able to develop IOT applications

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Explain the fundamentals and applications of IoT, its Architecture, Design Principles and Standards	1	
2	Apply programming skills to design IoT applications	3	
3	Analyze IoT system management	2	2
4	Design and Implement applications of IoT and make presentation in team	5,10	2

MODULE – 1	10 Hrs.

Introduction to Internet of Things: Definition and characteristics of IoT, Physical design of IoT, Things in IoT, IoT Protocols, Logical Design, IoT functional blocks, IoT communication Models, IoT communication API's, IoT enabling Technologies Wireless sensor networks, Cloud Computing, Big Data Analytics, Communication protocols, embedded systems.

IoT levels and deployment template Domain specific IoTs, - IoT levels, Introduction, Home Automation; Cities; Environment; Energy; Retail; Logistics; Agriculture; Industry; Health &Lifestyle.

MODULE – 2	10 Hrs.

IoT and M2M IoT System management with NETCONF-YANG Introduction, M2M, Difference between IoT and M2M, SDN and NFV for IoT- Software defined networking, network function virtualization Need for IoT Systems management; SNMP; Network Operator Requirements; NETCONF; YANG; IoT Systems management with NETCONFYANF; NETOPEER.

IoT platform Design Methodology - IoT Design Methodology; Introduction; Case Study on IoT System for Weather Monitoring.

MODULE -3	10 Hrs.

IoT Physical Devices and End points - What is an IoT device; Exemplary Device- Raspberry Pi, Linux on Raspberry Pi, Raspberry Pi Interfaces, Other IoT devices.

IoT Physical Servers & Cloud Offerings: Designing a Restful Web API, Amazon Web Services for IoT, AmazonEC2, Amazon Auto Scaling, AmazonS3, Amazon RDS.

MODULE -4	10 Hrs.

Case studies illustrating IoT Design: Introduction to IOT Design, Home Automation, Smart Lighting, Home Intrusion Detection, Cities, Smart Parking.

Data Analytics for IOT- Apache Hadoop, Using Hadoop Map Reduce for Batch Data Analysis.

Text Books:

1. Internet of Things - A Hands on Approach, ArshdeepBahga and Vijay Madisetti Universities Press, 2015

Reference Books:

1. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols, 2nd Edition, Wiley ISBN: 978-1-119-99435-0, 370 pages, January 2012.
2. Vijay Madisetti, ArshdeepBahga, Internet of Things: A Hands-On Approach Vijay Madisetti, 1st Edition ISBN-10: 0996025529, 2014

MOOC Course:

Design for Internet of things https://nptel.ac.in/courses/108/108/108108098/

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2			2											
CO3		3												2
CO4					3					3				3

Course Title	**OPTIMIZATION TECHNIQUES**		
Course Code	**22IS405C**	**L-T-P**	**(3-0-0)3**
Exam	**3 Hrs.**	**Hours/Week**	**3 Hrs**
SEE	**50 Marks**	**Total Hours**	**40**

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Apply problem solving techniques through OR approaches	2	-
2.	Formulate the problem using linear programming technique	1	-
3.	Analyze the optimal solution for the given problem by applying Transportation problems	3	-
4.	Analyze the strategies with different players through game theory approach.	3	-
5.	Analyze the sequence of jobs to be executed by machines for the given problem	3	-

MODULE – 1	10 Hrs.

Linear programming problem(LPP): Introduction, structure of linear programming model, advantages, general model of Linear programming problem(LPP), examples of LP formulation, graphical solutions of LP problem and Solution of LPP by simplex method:

MODULE – 2	10 Hrs.

Transportation and Assignment Problems: Mathematical model of transportation problem, methods of finding initial solution (Northwest corner rule, Least cost method, Vogel's approximation method), test for optimality in TP using MODI Method. Mathematical model of assignment problem, Hungarian method for solving assignment problem.

MODULE -3	10 Hrs.

Theory of games: Introduction, two-person zero sum games, pure strategies (MinMax and MaxMin principles), mixed strategies. The rules of principles of dominance, algebraic method to solve games without saddle point, graphical methods to solve games.

MODULE -4	10 Hrs.

Network Analysis: PERT and CPM, Network construction and determination of critical path, Calculation of ES, EF, LS, LF, TF, FF and IF, Crashing of a project, Scheduling of a project and resource levelling.

Text Books:

1. Operations Theory and Applications, J.K. Sharma, 5th edition, MacMillan publisher India(Chapter 1,2,3,4,5,910,11,12,20).
2. Operations Research S.D Sharma, Kedarnath, Ramnath and Co, 2002.

Reference Books:

1. Operations Research – An Introduction Taha H A- Low price edition 7th edition,2006.
2. Introduction to operation Research, Hiller and Liberman, Mc GRawHill , 5th edition ,2001.

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2													
CO2	3													
CO3			2											
CO4			2											
CO5			2											

Course Title	**PROBABILTIY, STASTISTICS AND QUEING**		
Course Code	**22IS405D**	**L-T-P**	**(3-0-0)3**
Exam	**3 Hrs.**	**Hours/Week**	**3 Hrs**
SEE	**50 Marks**	**Total Hours**	**40**

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1.	Understand the basics of probability, sample space, events, statistics and apply them to real life problems	1	-
2.	Distinguish probability density and distribution functions for single and multiple random variables.	1	-
3.	Use the probability, moment generating functions and characteristic functions.	1	-
4.	Formulate, analyze and validate models applicable to practical problems.	2	-

MODULE – 1	**10 Hrs.**
Probability Theory: Definition of probability: classical, empirical and axiomatic approach of probability, Addition theorem of	

probability, Multiplication theorem of probability, Bayes theorem of inverse probability, Properties of probabilities with proofs, Examples.

MODULE – 2	**10 Hrs.**

Random Variable and Mathematical Expectation: Definition of random variables, Probability distributions, Probability mass function, Probability density function, Mathematical expectation, Join and marginal probability distributions, Properties of expectation and variance with proofs, Examples.

MODULE -3	**10 Hrs.**

Correlation: Introduction, Types of correlation, Correlation and causation, Methods of studying correlation, Karl Pearson's correlation coefficient, Spearmans rank correlation, Coefficient, Properties of Karl Pearson's correlation coefficient, Properties of Spearmans rank correlation coefficient, Probable errors, Examples.

MODULE -4	**10 Hrs.**

Linear Regression Analysis: Introduction, Linear and non-linear regression, Lines of regression, Derivation of regression lines of y on x and x on y, Angle between the regression lines, Coefficients of regression, Theorems on regression coefficient, Properties of regression coefficient, Examples.

Text Books:

1. S. C. Gupta, "Fundamentals of Statistics", 46th Edition, Himalaya Publishing House.
2. G. V. Kumbhojkar, "Probability and Random Processes", 14th Edition, C. Jamnadas and co.

Reference Books:

1. Kishor S. Trivedi, "Probability, Statistics with Reliability, Queuing and Computer Science Applications", 2nd Edition, Wiley India Pvt. Ltd.
2. Vijay K. Rohatgi, A. K. Md. Ehsanes Saleh, An Introduction To Probability And Statistics, 3rd Edition, Wiley Publication.

Course Articulation Matrix

CO	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2	3													
CO3	3													
CO4		2												

Course Title	**Green IT and Sustainability**		
Course Code	**22IS406A**	**L-T-P**	**(2-0-0)1**
Exam	**3 Hrs.**	**Hours/Week**	**2 Hrs**
SEE	**50 Marks**	**Total Hours**	**20**

Course Objective: Students will be able to use appropriate data structures for solving problems.

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

#	**Course Outcomes**	**Mapping to PO's**	**Mapping to PSO's**
1.	Describe the concepts of how to manage the green IT with necessary components.	1	-
2.	Select hardware and software to facilitate more sustainable operation	2	-
3.	Relate the green computing practices to save energy	2,6,7	-
4.	Describe the use of IT in relation to environmental perspectives.	2,6,7	-

MODULE – 1	**5 Hrs.**

Green IT: **An Overview**

Green IT fundamentals - Environmental Impacts of IT - Green IT standards - Applying IT for enhancing environmental sustainability

MODULE – 2 **5 Hrs.**

Green Devices And Hardware

Life cycle of a device or hardware - Reuse, Recycle and dispose. Green software - Energy saving software techniques, Green information systems, evaluating software impact to platform power

MODULE -3 **5 Hrs.**

Managing Green IT

Implementation of Green IT, Information Assurance and communication - Green Enterprise transformation roadmap - Green compliance

MODULE -4 **5 Hrs.**

Law, Standards and Protocols

Regulatory environment and IT manufacturers, Non regulatory government initiatives, Green building standards, Green data centers.

Text Books:

1. Bhuvan Unhelkar, —"Green IT Strategies and Applications-Using Environmental Intelligence", CRC Press, June 2014

Reference Books:

1. Woody Leonhard, Katherine Murray, —Green Home computing for dummies, August 2012.
2. San Murugesan, G.R. Gangadharan"Harnessing Green IT Principles and Practices",Wiley Publication, ISBN:9788126539680.

Course Articulation Matrix

Course Outcomes	**Program Outcomes [POs]**													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2		2												
CO3		2				2	2							
CO4		2				2	2							

Course Title	**User Interface Design**		
Course Code	**22IS406B**	**(L-T-P)C**	**(1-0-0)1**
Exam	**3Hrs**	**Hours/Week**	**2**
SEE	**50 Marks**	**Total Hours**	**20**

Course Objective: Students will be able to apply the concepts and principles of User Interface Design and evaluate User Interfaces.

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

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	Describe motivations, theories and design processes of interactive systems	1	-
2	Appreciate the desirable features of good error messages, web pages, windows, documentation, help facility and visualization techniques	1	-
3	Analyze different types of user interfaces, devices and quality of service issues	2	-
4	Design appropriate user interface for given requirement	3	-

MODULE – 1	5Hrs.

Usability of Interactive Systems: Introduction, Usability Goals and Measures, Usability Motivations, Universal Usability, Goals for our Profession. Guidelines, Principles, and Theories: Introduction, Guidelines, Principles, Theories, four pillars of design.

MODULE – 2	5Hrs.

Direct Manipulation and Virtual Environments: Introduction, Examples of Direct Manipulation, Discussion of Direct Manipulation. Menu Selection, Form Fill-in, and Dialog Boxes: Introduction, Task Related Menu Organization, Single Menus, Combination of Multiple Menus.

MODULE -3	5Hrs.

Quality of Service: Introduction, Models of Response-Time Impacts, Expectations and Attitudes, User Productivity, Variability in Response Time, Frustrating Experiences.

MODULE -4	5Hrs.

User Documentation and Online Help: Introduction, Online Versus Paper Documentation, Reading from Paper versus from Displays, Shaping the Content of the Documentation, Accessing the Documentation, Online Tutorials.

Text Books:

1. Ben Schneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs - Designing the User Interface-Strategies for Effective Human-Computer Interaction, 5th Edition, Pearson Education Inc. - Dorling Kindersley (India) Pvt. Ltd., 2014

Reference Books:

1. Alan J Dix et. Al, —Human Computer Interaction, 3rd Edition, PHI, 2004
2. Wilber O Galitz, —The Essential Guide to User Interface Design - An Introduction to GUI Design, Principles and Techniques , Wiley Dreamtech India Pvt. Ltd., 2007

MOOC Course:

1. User Interface Design https://nptel.ac.in/noc/courses/noc19/SEM1/noc19-ar10/

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3													
CO2	3													
CO3		2												
CO4			2											

Course Title	INTRODUCTION TO WEB TECHNOLOGY

Course Code	**22IS406C**	**L-T-P**	**(0-0-1)1**
Exam	**3 Hrs.**	**Hours/Week**	**2**
SEE	**50 Marks**	**Total Hours**	**26**

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

#	Course Outcomes	Mapping to PO	Mapping to PSOs
1.	Understand the fundamentals of HTML, XHTML, Javascript, PHP and MYSQL	1	-
2.	Design programs using Javascript and PHP	2, 3	-
3.	Design and implement interactive Websites	3, 5	2

Initially Students will be taught the basic concepts about XHTML, Javascript, PHP, MYSQL and following programming exercises are carried out to understand the concepts.

1. Develop and demonstrate a XHTML document that illustrates the use of external style sheet, ordered list, table, borders, padding, color, and the <span> tag.
2. Develop and demonstrate, using Javascript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.
3. Using Javascript script, develop a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.
4. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
5. Write a PHP program to store current date-time in a COOKIE and display the 'Last visited on' date-time on the web page upon reopening of the same page.
6. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.
7. Using PHP and MySQL, develop a program to accept book information viz. Accession Number, Title, Authors, Edition and Publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

Text Books:

1. Robert W Sebesta. Programming the World wide web, 8th edition, Pearson Education

Reference Books:

1. Chris Bates. Web Programming building internet applications, 3rd edition, Wiley India
2. James Lee, Brent Ware. Open Source Web Development with LAMP, Pearson Education

Course Title	**TECHNICAL WRITING USING LATEX**

Course Articulation Matrix

Corse Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2		2	2											
CO3			3		3									3

Course Code	**22IS406D**	**L-T-P**	**(0-0-1)1**
Exam	**3 Hrs**	**Hours/Week**	**2 Hrs**
SEE	**50 marks**	**Total Hours**	**28 Hrs**

#	Course Outcomes	Mapping to PO	Mapping to PSOs
1.	Identify different types of technical documents for a given requirement.	1	-
2.	Create well-written short technical documents and typeset it in LaTex	5	-
3.	Create a simple report for a given technical topic, and typeset it in LaTex	10	-
4.	Create a sample research paper in a technical topic, and typset it in LaTex templates	10	-

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

MODULE-1 **7 Hrs**

Introduction: Introduction to LaTeX, its installation, and different IDEs. Creating the first document using LaTeX, organizing content into sections using article and book class of LaTeX.

MODULE-2 **7 Hrs**

Styling Pages" Reviewing different paper sizes, examining packages, formatting the page by setting margins, customizing header and footer, changing the page orientation, dividing the document into multiple columns, reading different types of error messages.

MODULE-3 **7 Hrs**

Formatting Content: Formatting text (styles, size, alignment), adding colors to text and entire page, and adding bullets and numbered items, writing complex mathematics.

MODULE-4 **7 Hrs**

Tables and Images: Creating basic tables, adding simple and dashed borders, merging rows and columns, and handling situations where a table exceeds the size of a page, adding an image, exploring different properties like rotate, scale.

Text Books:

1. Phillip A Laplante, “Technical Writing: A Practical Guide for Engineers and Scientists”, 1st Edition, CRC Press, 2011.
2. Tobias Oetiker, “The Not So Short Introduction to LATEX” 2e https://cslab.pepperdine.edu/warford/cosc320/lshort.pdf [retreived 18 Oct. 22]

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2					3									
CO3										3				
CO4										3				

Course Title	**BIOLOGY FOR ENGINEERS**		
Course Code	**22BEIS407**	**(L-T-P)C**	**(1-0-0)1**
Exam	**3 Hrs.**	**Hours/Week**	**2**
SEE	**50 Marks**	**Total Hours**	**28**

Course Objective: Realization of relation between Natural Engineering and man-made Engineering.
Course outcomes: At the end of course, student will be able to:

#	Course Outcomes	Mapping to PO's	Mapping to PSO's
1	To familiarize engineering students with basic biological concepts	1	
2	To involve students in an interdisciplinary vision of biology and engineering	2	
3	To gain an appreciation for how biological systems can be designed and engineered to substitute natural system	2	
4	To develop biological models using AI tools	3	

MODULE – 1	**6 Hrs.**

Introduction to Human Anatomy: Overview of human anatomy, Structural organization of the human body-cardiovascular system, endocrine system, digestive system, respiratory system, excretory system, lymphatic system, nervous system, muscular system and skeletal system.

MODULE – 2	**8 Hrs.**

Bioinspired Engineering based on human physiology: Circulatory system (artificial heart, pacemaker,stents), Nervous system (Artificial neural network).

MODULE -3	**8 Hrs.**

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

MODULE -4	**6 Hrs.**

Artificial Intelligence and Biology: Applications of AI in medical imaging, neural engineering, systems biology, microbiome and data mining.

Text Books:

1. Jenkins, C.H. Bioinspired Engineering, NY: Momentum press, 2012 ISBN: 97816066502259
2. A Practical Guide to Bio-inspired Design, Hashemi Farzaneh, Helena, Lindemann, Udo, Springer 2019, ISBN 978-3-662-57683-0

Course Articulation Matrix

Course Outcomes	Program Outcomes [POs]													
COs	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3													
CO2		3												
CO3		3												
CO4			2											

Course Title	**UNIVERSAL HUMAN VALUES**		
Course Code	**22UHV**	**(L-T-P)C**	**(0-1-0)1**
Exam	**3 Hrs**	**Hours/Week**	**2 Hrs.**
SEE	**50 marks**	**Total Hours**	**28 Hrs.**

Course Objective:

The course aims at development of value education by the right understanding through the process of self-exploration (about themselves), family, society and nature/existence. Strengthening of self-reflection by development of commitment and courage to act are presented as the prime focus throughout the course towards qualitative transformation in the life of the student.

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

COs	**Statement**	**POs**
CO1	Start exploring themselves, get comfortable with each other and with the teacher and they start appreciating the need and relevance for the course..Also they are able to note that the natural acceptance (intention) is always for living in harmony.	PO6,PO7, PO8, PO9, PO12
CO2	Differentiate between the characteristics and activities of different orders and study the mutual fulfillment among them and need to take appropriate steps to ensure right participation (in terms of nurturing, protection and right utilization) in the nature.	PO6,PO7, PO8, PO9, PO12
CO3	Present sustainable solutions to the problems in society and nature. They are also able to see that these solutions are practicable and draw roadmaps to achieve them.	PO6,PO7, PO8, PO9, PO12

MODULE-1 **8 Hrs**

Introduction to Value Education: Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Right Understanding, Relationship and Physical Facility, Happiness and Prosperity – Current Scenario, Method to Fulfill the Basic Human Aspirations

MODULE-2 **6 Hrs**

Harmony in the Human Being: Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self Lecture, Understanding Harmony in the Self Tutorial, Harmony of the Self with the Body to ensure self-regulation and Health.

MODULE-3 **8 Hrs**

Harmony in the Family, Nature and Existence: Harmony in the Family – the Basic Unit of Human Interaction, Values in Human-to-Human Relationship, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Understanding Harmony in the Society, Vision for the Universal Human Order.

Whole existence as Coexistence: Understanding the harmony in the Nature, Interconnectedness and mutual fulfilment among the four orders of nature recyclability and self-regulation in nature. Include practice sessions to discuss human being as cause of imbalance in nature (film “Home” can be used), pollution, depletion of resources and role of technology etc.

MODULE-4 **6 Hrs**

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

Self-Learning Activities-

1. Sharing about One self and Exploring Natural Acceptance
2. Exploring Harmony of Self with the Body
3. Exploring the Feeling of Respect
4. Exploring the Four Orders of Nature Lecture and Exploring Co-existence in Existence
5. Exploring Humanistic Models in Education, Exploring Steps of Transition towards Universal Human Order

Text Book and Teachers Manual-

1. The Textbook: A Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1
2. The Teacher's for a Foundation Course in Human Values and Professional Ethics, R R Gaur, R Asthana, G P Bagaria, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-53-2

Reference Books:

1. Jeevan Vidya: Ek Parichaya, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak,1999.
2. HumanValues, A.N.Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. The Story of My Experiments with Truth-by Mohandas Karamchand Gandhi
5. Small is Beautiful-E.F Schumacher.
6. Slow is Beautiful-Cecile Andrews
7. Economy of Permanence-JCKumarappa
8. Bharat Mein Angreji Raj–Pandit Sunderlal.
9. Redis covering India-by Dharampal
10. Hind Swarajor Indian Home Rule-by Mohandas K. Gandhi.
11. India Wins Freedom-Maulana Abdul Kalam Azad
12. Vivekananda-Romain Rolland(English)
13. Gandhi-Romain Rolland(English)

Course Articulation Matrix

Course Outcomes	**Program Outcomes [POs]**													
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1						2	1	3	2			1		
CO2						2	1	3	2			1		
CO3						2	1	3	2			1		