mLAC
ಶೀಲಂ ಪರಮ ಭೂಷಣಂ
Excellence in Education
Since 1972

# MAHARANI LAKSHMI AMMANNI COLLEGE FOR WOMEN (AUTONOMOUS)

Affiliated to Bangalore City University (BCU)

## BCA Programme (Basic and Honours degree)

## Syllabus for 1st to 4th Semesters

## Scheme Of Examination

## and
## Open Elective Courses in Computer Science

## BACHELOR OF COMPUTER APPLICATIONS (BCA)

**Choice Based Credit System (Semester Scheme)**
**Under National Education Policy NEP**

**Academic Year 2021-2022 onwards**

# DEPARTMENT OF COMPUTER SCIENCE

## Program Outcomes: BCA (3 Years) Degree

PO1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply design principles in the development of solutions for problems of varying complexity

PO2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyse problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.

PO3. **Design and Development of Solutions:** Ability to design and develop algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.

PO4. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tool or technique for modelling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.

PO5. **Communication:** Acquiring reasonably good communication skill both in oral and writing.

PO6. **Project Management:** Implementing existing projects and being able to improvise or launch own project by identifying gap in solutions.

PO7. **Ethics on Profession, Environment and Society:** Exhibiting Professional Ethics to maintain the integrity in a working environment and provide computer-based solutions for societal problems.

PO8. **Lifelong Learning:** The ongoing, voluntary, and self-motivated pursuit of knowledge for either personal or professional reasons helps in competitiveness and employability, and also enhances social inclusion, active citizenship, and personal development.

PO9. **Motivation to take up Higher Studies:** Inspiration to continue education towards advanced studies in Computer Science.

## Additional Program Outcomes: BCA Degree (Hons)

The Bachelor of Computer Application (BCA (Hons)) program enables students to attain following additional attributes besides the afore-mentioned attributes, by the time of graduation:

PO1. Apply standard Software Engineering practices and strategies in real -time software project development

PO2. Design and develop computer programs/computer -based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics.

PO3. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems

PO4. The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.

PO5. The ability to work independently on a substantial software project and as an effective team member.

# Program Specific Outcomes (PSO)

**PSO.1:** Understand nature, scope and application of computers and computer languages

**PSO.2:** Apply programming skills for software development

**PSO.3:** Effective communication skills

**PSO.4:** Critical thinking to understand the requirements of application development and implementation using any programming language or software tools

**PSO.5:** Specialization in Computer Science and Applications

**PSO.6:** Industry readiness

**PSO.7:** Research culture

**PSO.8:** Holistic personality development through value education, by organizing and participating in various activities

**PSO.9:** Entrepreneurship skills

**PSO.10:** Project development in latest technologies like Internet of Things, Machine Learning, Artificial Intelligence and Data Analytics

**PSO.11:** Understanding the environmental issues and sustainable development

**PSO.12:** Professional & Ethical responsibility

# BCA Curriculum Structure

## SEMESTER 1

| Course Type | Paper Code | Title of the paper | Instruction Hours /Week [L+T+P] | Duration of Exam (Hours) | Max. marks | | | Credits |
|---|---|---|---|---|---|---|---|---|
| | | | | | CIA | ESA | Total | |
| DSC | CA-C1T | Problem solving techniques using C | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C2T | Data Structures | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C3T | Discrete Structures | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C4L | C Programming Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| | CA-C5L | Data Structures lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| OE | OE1 | Journey into Fundamentals and C Programming Concepts | 3+0+2 | 2 | 40 | 60 | 100 | 3 |
| LANG | LANG1 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| | LANG2 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| SEC | Skill based | Digital Fluency / Office Management Tools | 1+0+2 | 2 | 20 | 30 | 50 | 2 |
| | Value based | Yoga | 0+0+2 | - | 25 | - | 25 | 1 |
| | | Health & Wellness | 0+0+2 | - | 25 | - | 25 | 1 |
| | | Total | | | | | 800 | 26 |

## SEMESTER 2

| Course Type | Paper Code | Title of the paper | Instruction Hours /Week [L+T+P] | Duration of Exam (Hours) | Max. marks | | | Credits |
|---|---|---|---|---|---|---|---|---|
| | | | | | CIA | ESA | Total | |
| DSC | CA-C6T | Object Oriented Programming using Java | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C7T | Database Management Systems | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C8T | Computer Architecture | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C9L | Java Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| | CA-C10L | DBMS Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| OE | OE2 | Python Programming | 3+0+2 | 2 | 40 | 60 | 100 | 3 |
| LANG | LANG1 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| | LANG2 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| AECC | AECC2 | Environmental Science | 2+0+0 | 2 | 20 | 30 | 50 | 2 |
| SEC | Value Based | Sports | 0+0+2 | - | 20 | - | 30 | 1 |
| | | NCC/NSS/R&R(S&G)/ Cultural | 0+0+2 | - | 20 | - | 30 | 1 |
| | | | | | | | 800 | 26 |

## SEMESTER 3

| Course Type | Paper Code | Title of the paper | Instruction Hours /Week [L+T+P] | Duration of Exam (Hours) | Max. marks | | | Credits |
|---|---|---|---|---|---|---|---|---|
| | | | | | CIA | ESA | Total | |
| DSC | CA-C11T | Operating Systems | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C12T | Computer Networks | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-C13T | Python Programming | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-14L | Computer Networks Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| | CA-15L | Python Programming Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| OE | OE3 | Fundamentals of HTML and CSS | 3+0+2 | 2 | 40 | 60 | 100 | 3 |
| LANG | LANG1 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| | LANG2 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| AECC | AECC3 | Indian Constitution | 2+0+0 | 2 | 25 | 25 | 50 | 2 |
| SEC | Skill Based | Artificial Intelligence/Computer assembly and Repair | 0+0+2 | 2 | 20 | 30 | 50 | 1 |
| | Value Based | Sports | 0+0+2 | - | 20 | - | 30 | 1 |
| | | NCC/NSS/R&R(S&G)/ Cultural | 0+0+2 | - | 20 | - | 30 | 1 |
| | | | | | | | 800 | 26 |

## SEMESTER 4

| Course Type | Paper Code | Title of the paper | Instruction Hours /Week [L+T+P] | Duration of Exam (Hours) | Max. marks | | | Credits |
|---|---|---|---|---|---|---|---|---|
| | | | | | CIA | ESA | Total | |
| DSC | CA-16T | Software Engineering | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-17T | Design and analysis of Algorithm | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-18T | Internet Technologies | 4+0+0 | 2 | 40 | 60 | 100 | 3 |
| | CA-19L | Design and analysis of Algorithm Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| | CA-20L | Internet Technologies Lab | 0+0+4 | 3 | 25 | 25 | 50 | 2 |
| OE | OE4 | Database Management Systems | 3+0+2 | 2 | 40 | 60 | 100 | 3 |
| LANG | LANG1 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | LANG2 | | 3+1+0 | 2 | 40 | 60 | 100 | 3 |
| **AECC** | AECC4 | Financial Education and Investment Awareness | 2+0+0 | 2 | 20 | 30 | 50 | 2 |
| **SEC** | Value Based | Sports | 0+0+2 | - | 20 | - | 30 | 1 |
| | | NCC/NSS/R&R(S&G)/ Cultural | 0+0+2 | - | 20 | - | 30 | 1 |
| | | | | | | | 800 | 26 |

# I SEMESTER BCA

## CA-C1T: PROBLEM SOLVING TECHNIQUES USING C

**Objective**: To develop procedure - oriented programming skills using C Language
**Skills to be developed:** Logical thinking, Problem Analysis, Algorithm Design, Implementation by Coding and Testing
**Course Outcomes:**
**CO1.1:** Introduction to software, programming concepts such as structured programming and modular programming, skill to develop logic for problems using algorithms and flowcharts and overview of C programming language
**CO1.2:** Managing input output operations, decision making structures, branching and looping structures with examples being implemented in C programming, declaring arrays and implementing single dimensional and two dimensional array problems
**CO1.3:** Ability to define data types and use them in simple data processing applications using the concept of strings, modular programming through function implementation, recursive functions, and use of pointers in C
**CO1.4:** Gain knowledge on storage classes, defining and using macros, ability to define structures and union, create text and binary files

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| **52** | **3** | **4** | **40** | **60** | **100** |

### Unit 1: Introduction [13 hours]

Software, Classification of Software, Modular Programming, Structured Programming, Role of algorithms in Computing, Algorithms as a Technology, Algorithms and Flowcharts with examples **Overview of C Language:** History of C, Character set, C tokens, Identifiers, Keywords, Data types, Variables, Constants, Symbolic constants, Operators in C, Hierarchy of Operators, Expressions, Type Conversions and Library Functions, Managing Input and Output Operations: Formatted and Unformatted I/O Functions

### Unit 2: Control Structures and Arrays [13 hours]

**Decision making, Branching and Looping:** Decision Making Statements, if Statement, if–else statement, nesting of if-else statements, else–if ladder, switch statement, Ternary operator, looping structures, while, do-while, for loop, Nested loops; break, continue, exchanging values of two variables, counting, summation of set of numbers, factorial computation, generating Fibonnaci series, reversing digits of integer, GCD of two numbers, finding square root of number, smallest divisor of integer, computing prime factors of an integer, character to number conversion, generation of pseudo random numbers, raising a number to large power

**Arrays:** Declaring and Initializing, One Dimensional Arrays, Two Dimensional Arrays, Multi- Dimensional Arrays, array order reversal, removal of duplicates from an ordered array, partitioning an array, finding kth smallest element, multiplication of matrices, Command Line Arguments

### Unit 3: Strings, Functions and Pointers [13 hours]

**Strings:** Declaring and Initializing strings, Operations on strings, Arrays of strings, text processing, keyword searching in text
**Functions:** Function Definition, prototyping, types of functions, passing arguments to functions, passing arrays to functions, passing strings to functions, Nested Functions, Call by value, Call by reference, Recursive functions.
**Pointers:** Declarations, Pointer arithmetic, Pointers and functions, Pointers and Arrays, Arrays of Pointers, Pointers and Structures, Meaning of static and dynamic memory allocation

### Unit 4: Storage Classes, Macros, Structures, Unions, Files [13 hours]

**Storage Classes:** Automatic, External, Static and Register Variables. **Macros:** Definition, types of Macros, Creating and implementing user defined header files. **Structures and Unions:** Declaring and Initializing, Nested structures, Array of Structures, Passing structures to functions, typedef, enum, Bit fields, Command Line arguments, C pre-processor directives. **Files:** File modes, File functions and File operations, Text and Binary file.

### TEXT BOOKS

1. R.G. Dromey, “How to solve it by computer”,Pearson Education India, 2008
2. E. Balaguruswamy, “Programming In ANSI C**”,** 5th edition**,** TMH Publications, 2011
3. Brain M Kernighan and Dennis M Ritchie,”The C Programming Language”, 2nd Edition, Princeton Hall Software Series, 2012
4. Ashok N. Kamthane, Amit Kamthane, “Programming in C”, Pearson Education,2015
5. Yeshwant Kanetkar, “Let Us C”, 14th Edition, BPB Publications, 2016

### REFERENCE BOOKS

1. Greg Perry and Dean Miller, “C Programming Absolute Beginner’s Guide”, 3rd Edition, Pearson Education, Inc, 2014
2. Byron Gottfried, “Programming in C”, Schaums’ Outline series, Third Edition, July 2017
3. Herbert Schildt : The Complete Reference -C”,4th Indian Edition, McGraw Hill, 2017

## CA-C2T: DATA STRUCTURES

**Objective:** To understand different types of data structures and their implementation

**Skills to be developed:** Designing, implementing and analysing algorithms

**Course Outcomes:**
**CO2.1**: Gain knowledge on Data organization, Data structures, design and analyze the time and space efficiency of the data structure, representation of linear arrays in memory, and operations on linear array, string processing and pattern matching algorithm, searching and sorting techniques implementation in C programming language.
**CO2.2:** Gain knowledge on linked list creation, operations on linked lists like insertion, deletion, types of linked lists -singly linked list, doubly linked list, circular linked list and their implementation using arrays and pointers and the concept of Garbage collection, implementation using C language.
**CO2.3**: Understanding stacks and queues, operations on stacks and queues**,** applications of stacks and queues – Recursion, Towers of Hanoi problem, Polish notation, implementation of stacks and queues

**CO2.4:** Understanding representation of graphs, traversing a graph, applying Depth First Search and Breadth First Search algorithms, Topological sorting; creation and traversal of Binary Search Trees, AVL and B trees, Lexicographic Search Trees

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

**Unit 1:** **[13 hours]**

**Introduction and Overview:** Definition, Elementary data organization, Data Structures, Operations on data structures, Abstract data types, Complexity of algorithms, Asymptotic notations
**Arrays:** Representation of Arrays in memory**,** Traversing Linear arrays, Insertion, and deletion, Sparse matrices
**String Processing:** Implementation of Strings using pointers, Word/text processing, Pattern matching algorithms
**Sorting:** Bubble sort, Insertion sort, Selection sort, Quick sort, Merge sort.
**Searching:** Linear Search, Binary search
**Unit 2:** **[13 hours]**

**Linked list:** Definition, Representation of Singly linked list in memory, singly linked list Operations-Traversing, Insertion, Deletion and Searching; Memory allocation functions, Garbage collection; Types of linked lists: Doubly linked list; Header linked list; Circular linked list, Implementation of Singly linked list, Doubly linked list, Circularly linked list

**Unit 3:** **[13 hours]**

**Stacks:** Definition, Array representation, Linked representation, Stack Operations; Applications of Stack: Recursion, Towers of Hanoi, Arithmetic Expressions: Polish Notation, Evaluation of Postfix expression, Conversion of Infix to Postfix expression

**Queues:** Definition, Array representation, Linked list representation, Queue Operations, Types of queues: Simple queue, Circular queue, Double ended queue, Priority queue, Applications of queues

**Unit 4:** **[13 hours]**

**Graphs:** Graph theory terminology; Sequential representation of Graphs, Adjacency matrix, Traversing a Graph- Depth First Search, Breadth First Search, Topological Sorting

**Binary Trees:** Definition, Properties, Types, Memory Representation, Creation, Binary Search Tree: Definition, Pre-order, In-order, Post Order Traversals, Searching, Insertion, and deletion, Height Balance AVL Trees, B Trees, Lexicographic Search Trees

**TEXTBOOKS:**

1. Robert Kruse, C.L.Tondo, Bruce Leung, Shashi Mogalla, “Data Structures and Program Design using C”, Pearson Education, 2009
2. E. Balaguruswamy, “Data Structures using C”, McGraw Hill Edition India Pvt. Ltd, 2013
3. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.

4. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.

### REFERENCES BOOKS:

1. Forouzan,"A Structured Programming approach using C",Second Edition, Pearson Education, 2013
2. Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
3. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Second Edition, Pearson Education, 2013
4. Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2 nd Ed, McGraw Hill, 2013
5. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014

## CA-C3T: Discrete Structures

**Objective:** To develop basic mathematical skills required for computer applications
**Skills to be developed:** proficiency in applying mathematical concepts to computational methods.
**Course Outcomes:**
**CO3.1:** Gaining knowledge on sets, set operations, functions and mathematical logic
**CO3.2:** Understanding the concept of counting and relations with examples.
**CO3.3:** Understand matrices, types, operations on matrices, applications of matrices to solve system of linear equations.
**CO3.4:** Gain knowledge on Graph terminologies, types of graphs, operations on graphs, incidence matrix, applications of trees and its connectivity.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | End Semester Exam | Total Marks |
|---|---|---|---|---|---|
| *52* | **3** | 4 | **40** | **60** | **100** |

### Unit 1: Set Theory, Functions and Logic [13 Hours]

**Set Theory:** Fundamentals of Set theory, Set Operations -Union, Intersection, Set difference and Complement of Sets, Types of sets – subsets, equal sets, universal sets, finite and infinite sets. Laws of Set Theory, Counting and Venn Diagrams, Cartesian Products
**Functions:** Domain and Range of function, Types of functions - One-to-One, Onto Functions, Function Composition and Inverse Functions, Mathematical Induction, The well ordering principle, Recursive Definitions, Structural Induction, Recursive algorithms
**Mathematical Logic:** Fundamentals of Logic, Propositional Logic, Logical Connectives and Truth Tables, Logic Equivalence, Predicates and Quantifiers

### Unit 2: Counting and Relations [13 Hours]

**Counting:** Basics of counting, Pigeonhole Principle, Permutations and Combinations, Binomial Coefficients, recurrence relations, Modelling with recurrence relations with examples of Fibonacci numbers and the Towers of Hanoi problem

**Relations:** Definition, Properties of relations, Types of Relations: Reflexive, Symmetric and Equivalence relations Divide and Conquer Relations with examples (No theorems), Representing relations using matrices and digraphs

### Unit 3: Matrices [13 Hours]

Matrices: Definition, order of a matrix, types of matrices, operations on matrices, determinant of a matrix, inverse of a matrix, rank of a matrix, linear transformations, applications of matrices to solve system of linear equations.

### Unit 4: Graph theory [13 hours]

**Graphs:** Introduction, Basic graph terminologies, Representation of graphs, Isomorphism of graphs, Operations on graphs, Planar graphs, Fundamentals of Digraphs, Matrix representation of graphs - Incidence matrix, circuit matrix, adjacency matrix and path matrix (no theorems)

**Trees:** Introduction, Application of trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees Prim's and Kruskal's Algorithms, Connectivity, Euler and Hamiltonian paths

**Textbooks:**

1) C. L. Liu, "Elements of Discrete Mathematics", Tata McGraw Hill, 2000
2) Ralph P Grimaldi, "Discrete and Combinatorial Mathematics", 5th Edition, Pearson Education, 2004
3) Richard Bronson, Schaums'Outline of Matrix Operations, McGraw Hill Publications, 2nd Edition, 2011
4) Dr. Grewal B.S., "Higher Engineering Mathematics", Khanna Publishers,43rd Edition, 2015
5) J P Tremblay, R Manohar, "Discrete Mathematical Structures with Applications to Computer Science", 2017
6) Narasingh Deo," Graph Theory with Applications to Engineering and Computer Science", Dover Publications Inc., New York,2017
7) Sheldon M Ross, "Introduction to Probability Models",12th Edition, Academic Press, 2019

**Reference Books**

1) Kenneth H Rosen, "Discrete Mathematics and its Applications", McGraw Hill Publications, 7th Edition, 2007
2) Sastry S.S, "Engineering Mathematics Volume 1", 4th Edition, PHI Learning, 2008
3) Peter V. O'Neil," Advanced Engineering Mathematics", 7th Edition, Cengage,2011

## CA-C4L: C PROGRAMMING LAB

**Course Outcomes:**

**COP4.1:** Ability to gain problem solving skills and implementation using C programming control structures such as decision making, looping, functions, structures and pointers.

**COP4.2:** Ability to gain problem solving and coding skills using C programming String manipulation, series generation, macros, command line arguments and files.

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | 25 | 25 | 50 |

### PART – A

1. Write a C program to find the roots of the given quadratic equation.
2. Write a C program to read percentage of marks and display appropriate grade.
3. Write a C program to find sum of the digits of a number, reverse the number and check for palindrome.
4. Write a C program to read numbers from keyboard continuously till user presses 999 and to find sum of only positive numbers.
5. Write a C program to remove duplicate element in a single dimensional array.
6. Write a C program to arrange the given set of numbers in ascending and descending order.
7. Write a C program to calculate Binomial Coefficient NCR= N! / (R! * (N-R)!) using recursion.
8. Write a C program to implement Call by value and Call by reference.
9. Write a C program to create a structure called STUDENT with the fields Register number, Name and Marks. Input details for 3 students. Display details of the student who has scored highest marks.
10. Write a C program to swap two numbers using pointers.

## PART – B

1. Write a C program to find the area of a triangle using area = √ s (s – a) (s – b) (s – c)
Where s = (a + b + c)/2.
2. Write a C program to enter an alphabet and print whether the given alphabet is a vowel or a consonant using switch case.
3. Write a C program to find GCD and LCM of 2 integers.
4. Write a C program to find whether a given number is prime or not.
5. Write a C program to generate Fibonacci sequence
6. Write a C program to print sum of series 1+1/2+1/3+1/4+….1/N
7. Write a C program to create a macro to find largest of three numbers.
8. Write a C program to illustrate command line arguments.
9. Write a C program to print multiplication of two matrices.
10. Write a C program to create a text file and print its contents.

**Scheme of Evaluation:**

| | |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05 Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |

## CA-C5L: DATA STRUCTURES LAB

**Course Outcomes:**

**COP5.1:** Implementation of string manipulation using pointers, searching techniques, sorting techniques, stacks, linear and circular queues, creation of binary search tree

**COP5.2:** Implementation of array operations, pattern matching, data structure operations on linked lists, applications of stack like recursion and Towers of Hanoi problem, evaluation of postfix expression

| Credits | Hours/Week | Internal | Exam | Total Marks |
|---|---|---|---|---|

|  |  | **Assessment Marks** |  |  |
|---|---|---|---|---|
| **2** | **4** | **25** | **25** | **50** |

### PART – A

1. Write a menu driven C program to perform the following string operations using pointers (i) String Length (ii) String Concatenation(iii) string reverse
2. Write a program to find the location of a given number using Linear search.
3. Write a C program to search for an element in an array using Binary search.
4. Write a C program to sort a list of N elements using Selection Sort Algorithm.
5. Write a C program to construct a singly linked list and perform insertion at the beginning, deletion at the specified position and display the elements.
6. Write a C program to demonstrate the working of stack using arrays.
7. Write a C program for Towers of Hanoi problem using Recursion.
8. Write a C program to convert infix arithmetic expression to postfix expression.
9. Write a C program to simulate the working of Circular Queue using an array.
10. Write a C program to create and traverse a binary search tree.

### PART – B

1. Write a C program to demonstrate the following operations in an array
   (i) insertion of an element (ii) deletion of an element (iii) traversal
2. Write a C Program to copy one string into another without using built-in string functions.
3. Write a C program to find a pattern in a given text using pattern matching algorithm.
4. Write a C program to sort a list of elements using Bubble sort algorithm
5. Write a C program to sort a list of N elements using Insertion Sort Algorithm.
6. Write a C program to simulate the working of Linear Queue using linked list.
7. Write a C program to count the number of nodes in a linked list.
8. Write a C program to search for a given element in a sorted linked list.
9. Write a C program for evaluation of postfix expression.
10. Write a C program to implement stack operations using Linked list.
11. Write a C program to create a directed graph using Adjacency Matrix.

## Scheme of Evaluation:

|  |  |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05 Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |

# II SEMESTER BCA

## CA-C6T: OBJECT ORIENTED PROGRAMMING USING JAVA

**Objective:** To understand the concepts of object-oriented programming and Java programming features

**Skills to be developed:** Developing and implementing object-oriented programming using Java

**Course Outcomes:**

**CO6.1:** Understanding the concepts of Object-oriented programming and basics of JAVA programming

**CO6.2:** Understanding inheritance, polymorphism, Interfaces and Packages, and Exception handling in JAVA

**CO6.3:** Gain knowledge on Event and GUI programming, Applets, AWT, Swings, I/O Programming, Streams and Files

**CO6.4**: Gaining knowledge on Multithreading, Synchronization and Java Collections framework

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | End Semester Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

**Unit 1** **[13 hours]**

**Introduction to Object oriented programming**: Definition of OOPs, Basic concepts of OOPs, Objects and Classes; Comparison between Procedure oriented and objected oriented languages

**Basics of Java programming:** Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in Java. Objects and Classes: Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference

### Unit 2 [13 Hours]

**Inheritance and Polymorphism:** Inheritance in Java, Super and sub class, Overriding, Final variables and methods, Finalizer methods, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract methods and classes, Interface in Java, Package in Java, UTIL package.

**Managing Exceptions:** Introduction, Types of Exception Handling Code, Multiple Catch Statements, Using Finally Statement, Throwing Our Own Exceptions, Using Exceptions for Debugging.

### Unit 3 [13 Hours ]

**Event and GUI programming:** Event handling in Java, Event types, Mouse and key events, Delegation Event model, Event classes, Event listener interfaces, AWT, AWT classes, AWT controls, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Creating an Executable applet, running the Applet, Getting Input from the User, Introduction to Swing,

**I/O programming:** Managing Input/Output files in Java, Concept of Streams, Stream Classes: Byte Stream Classes, Character Stream Classes, Using the File Class, Creation of Files, Reading /Writing Characters, Reading / Writing Bytes, Collections in Java, Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.

### Unit 4 [13 Hours ]

**Multithreaded Programming:** Introduction, Creating Threads, Extending the Thread Class, Life Cycle of a thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the 'Runnable' Interface.

**Collections in Java:** Set, List, Queue, Deque Interfaces and ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet classes

**Introduction to JavaBeans and Network Programming**

**TEXTBOOK:**

1. E. Balaguruswamy, "Programming with JAVA: A Primer", 3rd Edition, Tata McGraw Hill, 2014

**REFERENCE BOOK:**

1. Raj Kumar Buyya, Object Oriented Programming with JAVA, McGraw Hill, 2009

2. Herbert Schildt," JAVA : The Complete Reference", 8th Edition, McGraw Hill, 2011

## CA-C7T: DATA BASE MANAGEMENT SYSTEMS

**Objective:** To understand database and database management concepts

**Skills to be developed:** Creating, modifying and maintaining databases, effectively querying databases to retrieve data as required.

**Course Outcomes:**

**CO7.1:** Understanding Database Management System, Database concepts, architecture, classification of DBMS.

**CO7.2:** Gain knowledge on Data modelling using ER model, data model, Record storage and primary file organization, Hashing techniques.

**CO7.3:** Relational Data Model and Relational Algebra, Examples of Relational Algebra queries, Relational Database Language, Data types, DDL and DML queries, Nested Queries, Applications of various types of SQL queries and their implementation

**CO7.4:** Functional Dependencies, Transitive Dependency and Normalization for Relational Database, Understand Transaction Processing Concepts, Concurrency Control techniques, Distributed Databases and Client Server Architecture and ACID properties.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

## Unit 1 [13 Hours]

**Introduction:** Database, Database management System, Database System; Characteristics of the Database Approach, Different people behind DBMS, Uses of DBMS, Implications of Database Approach, when not to use a DBMS;

**Database System Concepts and Architecture:** Data Models, Schemas and Instances, DBMS Architecture and Data Independence, Database languages and interfaces, The database system Environment; Centralized and Client-Server Architectures, Classification of DBMS.

## Unit 2 [13 Hours]

**Data Modelling Using the Entity-Relationship Model:** High level conceptual Data Models for Database Design with example; ER model concepts: Entities and Attributes, Entity types, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Relationships and roles, Notation for ER Diagrams;

**Record Storage and Primary File Organization:** Secondary Storage Devices, Buffering of Blocks, Placing file Records on Disk; Operations on Files: Files of unordered Records (Heap files), Files of Ordered Records (Sorted files), Indexing and its types

## Unit 3 [13 Hours]

**Relational Data Model and Relational Algebra:** Concepts, Constraints and Update operations, Defining Relations; Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from SET Theory- UNION, INTERSECTION, SET DIFFERENCE, Binary Relational Operations: JOIN and DIVISION Examples of queries in the Relational Algebra;

**Relational Database Language:** Data definition in SQL, Queries in SQL, Insert, Delete and Update Statements in SQL, Views in SQL, Specifying General Constraints as Assertions, Specifying indexes, Embedded SQL;

## Unit 4 [13 Hours]

**Relational Database Design:** Anomalies in a database, Functional Dependencies, Transitive dependencies and Normalization for Relational Database, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Lossless Join and dependency, Boyce-Codd Normal Form, higher order Normal Forms

**Transaction Processing Concepts:** Introduction, Transaction and System Concepts, Desirable properties of transaction, Schedules and Recoverability, Serializability of Schedules

**Concurrency Control techniques:** Locking Techniques, Concurrency Control based on time stamp ordering, Optimistic Concurrency control techniques

**Distributed Databases and Client Server Architecture:** Introduction, Overview of Client-Server Architecture, Data fragmentation, Replication and Allocation techniques, Types of Distributed database systems

**Textbook:**

1. Remez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", 7th Edition, Addison Wesley, 2016.
2. Silberschatz, Korth and Sudharshan, "Database System Concepts", 7th Edition, Tata McGraw Hill,2019

**References:**

1. C.J.Date, "Introduction to database systems", Eight Edition, Addison Wesley, 2004
2. Nilesh Shah, "Database Systems Using Oracle", 2nd Edition,|Pearson Education,2004

## CA-C8T: COMPUTER ARCHITECTURE

**Objective:** To impart knowledge on Computer Organization, logic design and architecture

**Skills to be developed:** To design digital logic circuits

**Course Outcomes:**

**CO8.1:** Gaining knowledge on data representation, Computer arithmetic, basics of computer architecture and organization, Digital logic circuits

**CO8.2:** Gain knowledge on basic computer organization and design and various computer instructions, processor structure and addressing modes

**CO8.3:** Understanding register transfer, microoperations, Input-Output organization and parallelism

**CO8.4:** Understand the memory organization, memory systems, mapping process, external memory and external interconnection standards.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | End Semester Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

**UNIT-1** **[13 Hours]**

**Data representation:** Fixed- and Floating-point numbers, Number Systems: Binary, Octal, Hexa decimal numbers, Inter conversions, One's and Two's complements, Positive and Negative numbers, Digital

Codes – ASCII, EBCDIC, Unicode, BCD, XS3, Gray Codes **Computer Arithmetic:** Addition and Subtraction Binary numbers, Multiplication and Division algorithms, Floating-point Arithmetic Operations, Decimal arithmetic operations

**Overview of Computer Organization**: Computer types, Functional units, Basic operational concepts, performance, Computer Architecture – Definition, Types, Von-Neumann Architecture, Bus Structures, Software, Performance, Multiprocessors and Multicomputer; **Digital Logic Circuits:** Logic gates, Boolean algebra, Map Simplification, Combinational circuits: Half Adder, Full Adder, Parallel Binary Adder, Sequential circuits: Flip flops, Shift Registers, Counters; Integrated Circuits, Multiplexer, Demultipexer, Encoder, Decoder, Parity Generators/Checkers,

## UNIT-2: [13 Hours]

**Basic Computer Organization and Design:** Instruction codes, Computer Registers, Computer Instructions and Instruction cycle, Timing and Control, Types of Instructions: Register reference, Memory-Reference, Input-Output reference, Instruction Cycle, Interrupt Cycle, Timing and Control

**Central Processing Unit:** Stack organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Complex Instruction Set Computer (CISC), Reduced Instruction Set Computer (RISC), CISC vs RISC

## UNIT-3 [13 Hours]

**Register Transfer and Microoperations:** Register Transfer Language, Register Transfers, Bus and Memory Transfer, Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic Logic Shift Unit

**Micro-programmed Control:** Control Memory, Address Sequencing, Micro-Program example, Design of Control Unit

**Input Output:** I/O interface, Programmed IO, Memory Mapped IO, Interrupt Driven IO, Direct Memory Access DMA, I/O Channels and processors

**Instruction level parallelism:** Instruction level parallelism (ILP), Overcoming data hazards, Limitations of ILP, Types of Parallel Processor System – SISD, SIMD, MISD, MIMD, Multiprocessors and Thread Level parallelism

## UNIT-4 [13 Hours]

**Memory System -** Basic Concepts, Memory Hierarchy, Semiconductor Random Access Memories RAM, Read-only Memories ROM, Types of ROM, Cache Memory – Levels of Cache, Performance Considerations-Hit Rate and Miss Penalty, Memory Connection to CPU, Associative Memory, Virtual Memory, External Memory – Magnetic Disk, Optical Memory, Solid State Drives, RAID

**External Interconnection Standards** – USB, SCSI, PCI Express, SATA.

## TEXTBOOK:

1. M. Morris Mano, "Computer System, Architecture", 2nd Edition, Pearson Education,2007
2. Kai Hwang "Advanced Computer Architecture-Parallelism, Scalability, Programmability", Tata McGraw Hill 2008
3. William Stallings, "Computer Organization & Architecture Designing for Performance", Tenth Edition, Pearson Education, 2016.

## REFERENCE BOOKS:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Naraig Manjikian, “Computer Organization and Embedded Systems”, Sixth Edition, McGrawHill, 2002.
2. Andrew S Tanenbaum “Structured Computer Organization”, 5th Edition, Pearson Education Inc 2006
3. Ronald J Tocci, Neal S Widmer, Gregory L Moss, “Digital Systems”, Tenth Edition, Pearson Education, 2009.
4. Subrata Ghoshal, “Computer Architecture and Organization”, Pearson India 2011
5. Morris Mano, “Digital Design”, 5th Edition, Prentice Hall, 2013
6. Thomas L. Floyd, ’Digital Fundamentals”, Pearson Education Inc, New Delhi, 2014

## CA-C9L: JAVA PROGRAMMING LAB

**Course Outcomes:**

**COP9.1:** Ability to gain programming skills by implementing the concepts of string operations, method overloading, method overriding, constructor overloading, multi-threading and file handling in Java.

**COP9.2:** Implementation of inheritance, Exception handling, Applet programming, AWT concepts, Event handling and Animation in Java.

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | 25 | 25 | 50 |

### PART – A

1. Write a program to find factorial of list of number reading input as command line argument.
2. Write a program to display all prime numbers between two limits.
3. Write a program to check all math class functions.
4. Write a program to insert an element in an existing array.
5. Write a program to implement all string operations.
6. Write a program to find area of geometrical figures using method.
7. Write a program to calculate bonus for different departments using method overriding.
8. Write a program to implement constructor overloading by passing different number of parameters of different types.
9. Write a program to implement thread priorities.
10. Write a program to copy a file to another.

### PART - B

11. Write a program to implement derived class constructors with ‘super’ keyword.
12. Write a program to sort list of elements in ascending and descending order and show the exception handling.
13. Write a menu driven program to display reverse of a string and to append a string.

14. Write a program to implement static data members and static member methods using classes.
15. Write a program to implement applets using AWT components.
16. Write a program to create student report using applet, read the input using text boxes and display the o/p using buttons.
17. Write a program to implement mouse events.
18. Write a program to implement keyboard events.
19. Write a program to implement thread, applets and graphics by implementing animation of ball moving.
20. Write a program to find whether the file exist or not in the specified directory.

### Scheme of Evaluation is as follows:

| | |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |

## CA-C10L: DATABASE MANAGEMENT SYSTEM LAB

**Course Outcomes:**

**COP10.1:** Design of queries for creation of database, tables with primary key and foreign key constraints, data manipulation operations like insertion, deletion, modification, creation of views and displaying of records

**COP10.2:** Creation of multiple tables in database, establishing relationship between them and designing complex queries, nested queries, Entity Relationship diagram for case studies of Bank Database and College Database

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | 25 | 25 | 50 |

**PART - A**

1. Write DDL queries for the following:
   i. Create database COLLEGE
   ii. Create the table STUDENT in the COLLEGE database with the following fields:
   regno: int, name: string, dob: date, address: string, email: string with regno as primary key
   iii. Add a new attribute phoneno to the existing table.
   iv. Change the data type of regno from integer to string.
   v. Delete the field email from the table
   vi. Rename the table as CANDIDATE
   vii. Drop the table CANDIDATE

2. A LIBRARY database has a table with the following attributes.

BOOK (bookid:string,title:string,author:string,publisher:string;yearpub:date,cost:number(5,2)

i. Create the table BOOK with bookid as primary key.
ii. Enter five tuples into the table.
iii. Display all the tuples in BOOK table.
iv. Display all the publishers.
v. Display distinct publishers.
vi. Arrange the tuples in the alphabetical order of the book titles.
vii. List the details of all the books whose cost lies in a specific range.

3. The COMPANY database of an organization has tables EMPLOYEE, DEPARTMENT and SALARY
EMPLOYEE (empid: string, empname:string, dob:date, designation:string,dno:string) with empid as primary key
DEPARTMENT (dno:string;dname:string) with dno as primary key
SALARY (empid:string,dno:string,salmonth:string,salary:number(8,2)) with composite key as empid and salmonth.

i. Create the above tables.
ii. Enter three tuples into each of the tables.
iii. Display the number of employees working in each department.
iv. Find the sum and average of the salaries of employees of a particular department for a specific month.
v. Find the least and highest salaries that an employee draws in a particular month.

4. Consider the VEHICLE_INSURANCE database given below. The primary keys are underlined and the data types are specified.
POLICYHOLDER (policyid:string, dlno: string, name: string, address: string, phone:number(10))
ACCIDENT (reportno: string, policyid:string, accdate: date, dlno:string, vregno:string, location: string, damageamt:number(10))

i. Create the above tables by properly specifying the primary keys and the foreign keys .
ii. Enter at least five tuples for each relation.
iii. Find total number of times damageamt is claimed in a specific year by a policy holder.
iv. List all accidents that occurred in a specific month.

5. Create a database UNIVERSITY which has the following tables COURSE, STUDENT, BOOK, ADOPTION for each course.
COURSE (courseid:string, coursename:string)
STUDENT (regno: string, name: string, courseid: string)
BOOK (bookid:string, booktitle: string, publisher: string, author: string)
ADOPTION (courseid: string, sem: string, bookid: string, regno:string)

i. Create the above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.
iii. Create a view to display the details of all the books adopted by a particular student in a specific semester.
iv. Count the number of books adopted by a particular student in a specific semester.

6. The following tables are maintained by a BOOK DEALER
AUTHOR (authorid: string, name: string, address: string, phone: number (10))
PUBLISHER (pubid: string, pubname:string, pubaddress:string, pubphone:number(10))
CATALOG (bookid: string, title: string, authorid: string, pubid: string, yearofpub: int, price: int)
ORDERDETAILS (orderno: int, bookid:string , quantity: int)

i. Create above tables by properly specifying the primary keys and the foreign keys.
ii. Enter at least five tuples for each relation.

iii. Display all the records of CATALOG table.
iv. List all the books published in a specific year.
v. Create view PRESALES to display the title and price of books of a particular author.
vi. Demonstrate how the price of books listed in view PRESALES is increased by 10%.

7. Write a PL/SQL program to find the sum of first 10 natural numbers using loop structure.

8. Write PL/SQL program to multiply two numbers using a function.

## PART – B

1. Create a database COMPANY which has entities EMPLOYEE, DEPARTMENT and PROJECT.
An employee has the fields: empid, empname, dob, doj, address and phone
A department has the fields: deptno, deptname, location
The employee works on a project which has the fields: projid, projname, completiondate, location.

    i. Create the above tables with primary key.
    ii. Insert at least 5 tuples.
    iii. Display all the tuples of the three tables.
    iv. Display all the employees in alphabetical order of empname.
    v. Display all projects in descending order of completiondate.
    vi. Display all the projects undertaken in a specific location of a particular department.

2. Create the following tables specifying the primary keys and foreign keys.
DEPARTMENT (deptid, deptname)
STUDENT (studid, studname, studaddr, deptid,sem)
BOOK (bookid, bookname, authorid, publisher, deptid)
AUTHOR (authorid, authorname, country)
BORROW (studid, bookid, borroweddate)

    Also write SQL queries for the following:
    i. List the details of students who are in second semester BCA.
    ii. List the students who have not borrowed any books.
    iii. Display the studid,studname,deptname,bookname,authorname,borroweddate of second semester BCA students who borrowed books.
    iv. Display the number of books written by each author.

3. Create the following tables specifying the primary keys and foreign keys. Also write SQL queries for the following:
DATABASE NAME: INVENTORY
PRODUCT (pid:string, pname:string)
SUPPLIER (supplier_id:string, supplier_name:string, pid: string)
PURCHASE_ORDER (p_order_id:string; supplier_id:string; total_amount:number(6,2); purchase_date:Date)
PURCHASE_DETAILS (p_details_id:string, p_order_id:string, product_id:string, unit_price:numeric(6,2), quantity:numeric(6,2))

    i. Create the above tables with primary key.
    ii. Insert at least 5 tuples.
    iii. Display all the tuples in alphabetical order of productname.
    iv. Display all the tuples in descending order of purchase_date.
    v. Display the product with least quantity.

vi. Display the supply who supplies a specific product.

4. Create a database COLLEGE1 which has entities STUDENT, COURSE, and ENROLL
STUDENT(sid:string,sname:string,address:string,phone:number(10),age:int)
COURSE (cid:string,cname:string)
ENROLL(sid:string,cid:string)

i. Create the above tables with primary key.
ii. Insert at least 5 tuples.
iii. Display all the tuples of the three tables.
iv. Display all the students who have enrolled for either the course DS or DBMS

5. Create a Database SETDB with the table names: Student, Instructor

STUDENT (id: int, name: string)
INSTRUCTOR (id: int, name: string)

i. Create the above tables and add at least 5 tuples.
ii. Display the tuples for those who are either students or instructors.
iii. Display the tuples for those who are both student and instructor.
iv. Display the tuples for those who are students but not instructors.
v. Display the tuples for those who are instructors but not students.

6. Create a Database Name: FACTORY with the table names: DEPARTMENT, EMPLOYEE

DEPARTMENT (dno: int, dname: string;)
EMPLOYEE (empid: string, empname: string, dno: int)

i. Create the above tables with primary and foreign keys.
ii. Add at least 5 tuples into each table.
iii. Display the names of managers of each department.
iv. Illustrate left join, right join and full outer join operations.

7. Create an ER diagram for the Bank Database.
Identify the entities, relationships, key attributes, other relevant attributes for the following specifications.
i. A bank has many branches in a country.
ii. Each branch is identified with the help of branch-id.
iii. Each branch of the bank has multiple customers.
iv. Customers also include employees of that particular branch.
v. An employee can be a manager or staff of that branch.
vi. Branch gives each customer an id in order to identify them.
vii. Customers can open two types of accounts i.e. Savings account and Current account. Branches of the bank provide Loan to the customers if required.
viii. Re-Payment of loan is done with the help of payment number of that particular loan.

8. Design an Entity Relationship (ER) model for a College database. Identify the entities, relationships, key attributes, other relevant attributes for the following specifications.
i. A college contains many departments.
ii. Each department can offer any number of courses.
iii. Many instructors can work in a department.
iv. An instructor can work only in one department.

v. For each department there is a Head.
vi. An instructor can be head of only one department.
vii. Each instructor can take any number of courses.
viii. A course can be taken by only one instructor.
ix. A student can enroll for any number of courses.
x. Each course can have any number of students.

### Scheme of Evaluation:

| | |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05 Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |

# III SEMESTER

# CA-C11T: OPERATING SYSTEMS

**Objective:** To introduce the fundamentals and major concepts of operating systems and operations of various components of an operating system.

**Skills developed:** ability to schedule processes and extend OS services using system calls

**Course Outcomes:**

**CO13.1:** Gain Knowledge on basic Operating System concepts, add process relevant to Operating System such as process creation, management, and scheduling algorithms.

**CO13.2:** Gain knowledge on process synchronization, process state transitions etc

**CO13.3:** Understanding the concepts of Memory Management, Page Management, File Management in Operating System

**CO13.4:** Gain knowledge on protection and security of Operating System

| Total Number Of Hours | Credits | Hours/Wee k | Internal Assessment Marks | End Semester Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

## UNIT - I

**Introduction:** Batch Systems, Concept of Multiprogramming and Time Sharing Parallel, Distributed, and real time System, Operating System Structure: Components and Services, System Calls, System Programs, Virtual machines; **Process Management**: Process Concept, Process Scheduling, Processes and Threads, Inter process communication, CPU Scheduling Criteria, Scheduling algorithm, Multiple Processor Scheduling, Real time Scheduling, Algorithm evolution.

[13 Hours]

## UNIT - II

**Process Synchronization:** The Critical Section Problem, Synchronization hardware, Semaphores, Thread Synchronization, Producers – Consumers Problem, Readers-writers problem; **Deadlocks**: System model, Characterization, Deadlock Prevention, Avoidance and detection, Recovery from deadlock, Distributed deadlocks, Combined approach to deadlock handling.

[13 Hours]

## UNIT - III

**Memory Management:** Logical and Physical address space, Address translation in paging system, Contiguous allocation, Paging and Segmentation; **Virtual Memory:** Page replacement algorithms, Allocation of frames, thrashing, Virtual memory addressing in memory management.

[13 Hours]

## UNIT – IV

**File Management** File Concepts, Access methods, Directory Structure, Protection, and consistency. File System Structure: Free space management, Directory Implementation, Efficiency and Performance, Recovery; Disk Structure & Scheduling methods, Disk management.

**Protection and Security:** Goals of protection, Domain Protection, Access matrix, Security Problem, Authentication, program threats, System Access Threats.
[13 Hours]

## Text Books:

1. Silberschatz and Galvin, "Operating System Concepts", 9th Edition, Wiley, 2012

## Reference Books:

1. H. M. Deitel, "Operating Systems", 3rd Edition, Prentice Hall, 2003
2. William Stallings, "Operating Systems", 6th Edition, Pearson Education, 2010

# CA-C12T: COMPUTER NETWORKS

**Objective:** To understand data communication concepts, Computer networks, protocols and standards
**Skills to be developed:** Understanding network models, communication protocols and implementing for organization and individuals.
**Course Outcomes:**
**CO12.1:** Ability to understand Data communication concepts, Computer Networks, protocols, standard model of communication (OSI and TCP/IP) and Physical and data link layer.
**CO12.2:** Ability to understand the concept of data link control and MAC, polling, reservation and token passing.
**CO12.3:** Ability to understand the concept of network layer and its protocol, routing concepts and to have knowledge on next generation IP address.
**CO12.4:** To have knowledge on transport layer and its protocol, UDP, TCP and application layer protocols like Domain Name System; Telnet, E-mail, FTP, WWW and HTTP.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

## UNIT – I

**Introduction:** Data Communications, Networks, Network Types, Internet History, Network Models: Protocol Layering, The OSI Model, TCP/IP Protocol Suite, **Introduction to Physical Layer:** Transmission Impairments, Data Rate Limits, Performance; **Multiplexing**: Frequency Division, Wavelength Division, Synchronous Time-Division, Statistical Time-Division, Spread Spectrum; Guided Media and unguided media. [13 hours]

## UNIT – II

**Introduction to Data-Link- Layer:** Link-Layer Addressing, Error Detection and Correction: Block Coding, Cyclic Codes, Checksum; **Data Link Control:** Data-Link Layer Protocols, HDLC, Point-To-Point (PPP),

**Media Access Control (MAC):** ALOHA, CSMA, CSMA/CD, CSMA/CA, Reservation, Polling, Token Passing, FDMA, TDMA, CDMA. [13 hours]

### UNIT – III

**Introduction to Network Layer:** Network-Layer Services, Packet Switching, Network-Layer Performance, IPV4 Addresses, **Network Layer Protocols:** Internet Protocol (IP), ICMPv4, Mobile IP, **Unicast Routing:** Routing Algorithms, Unicast Routing Protocols, **Next Generation IP:** IPv6 Addressing **Mapping:** ARP, ICMP, IGMP; Delivery, Forwarding, Routing: Unicast and Multicast.

[13 hours]

### UNIT – IV

**Introduction to Transport Layer:** Introduction, Transport-Layer Protocols, **Transport-Layer Protocols:** User Datagram Protocol, **Transmission Control Protocol:** TCP Services, TCP Features, Segment, A TCP Connection, TCP Congestion Control, Flow Control, Error Control, **Application Layer:** WWW, E-MAIL, Domain Name System (DNS), **Quality of Service:** Flow Control To Improves QoS, Integrated Services [13 hours]

### Text Books:

1. Behrouz A Forouzan, "Data Communications and Networking", Fifth E=dition, Mc Graw Hill Education, 2017

### Reference Books:

1. Andrew S. Tanenbaum, David J. Wetherall, "Computer Networks", 5th Edition, Prentice Hall,2011.
2. Larry L. Peterson and Bruce S. Davie, "Computer Networks A System Approach", 5th Edition,MKP, 2012.
3. James F. Kurose , Keith W. Ross, " Computer Networking, A Top-Down Approach", 5thEdition, Pearson, 2012

## CA-C13T: PYTHON PROGRAMMING

**Objective:** To introduce core programming basics – including data types, control structures, algorithm development, and program design with functions – via the python programming language. **Skills to be developed:** This course helps students in coding using object-oriented programming.

**Course Outcomes:**

**CO13.1:** Understand Python programming basics and control structures.

**CO13.2:** Gain knowledge on use of Python functions, anonymous functions, recursive functions and modules for code reusability, command line arguments and to be able to write simple Python programs for solving problems.

**CO13.3:** Ability to use data structures in Python such as arrays, lists, tuples, dictionaries and sets.

**CO13.4:** Ability to develop programmes on files, Binary data interchange formats, Exception handling, defining and using regular expressions in Python, small applications on GUI, Data Visualization, Database connectivity in Python.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | End Semester Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

### Unit I

**Introduction:** Introduction to Python programming, writing a Python program, Python Interactive Shell, Building blocks – Values and Variables, Variables and Assignment, Identifiers, Expressions and Arithmetic, Operators and operands, Order of operations, Operations on strings, Control codes within strings, String formatting, String operations, Composition, Comments, User input and Output, Errors in Python: syntax errors, run-time exceptions, and logic error.

**Control Structures:** Conditional expressions, Simple IF statement, IF...ELSE statement, Compound Boolean expressions, pass statement, Nested conditionals, Multi-way Decision statements, Iteration statement – While, for statement, nested loops, break and continue statements, while/else and for/else statements [13 hours]

### Unit II

**Functions and Modules:** Function calls, Built-in Functions, Standard Mathematical functions, Random functions, Time functions, system specific functions, eval and exec functions, String functions, writing functions- Global variables, Default parameters, importing a module using import statement, Functions as Data, return statement, Anonymous function using Lambda expression, Generators, Local function definitions, Recursion, Command Line Arguments;

**Objects and Classes:** Using objects, String objects, File objects, Fraction objects, Turtle graphics objects, Abstract Data types and classes, Inheritance [13 hours]

### Unit III

**Lists:** Using Lists, Building lists, List length, List membership, List operations: Traversal, Element Removal, List Slicing, List element removal, Lists and Functions, Cloning lists, List methods, Multidimensional lists, Nested lists, Comparison between Lists and Generators, Sparse matrices;

**Tuples:** Tuple assignment, Tuples as return values, Mutability and tuples, Comparison between Lists and Tuples;

**Dictionaries:** Using, Counting, Grouping, Keyword Arguments, Dictionary operations, Dictionary methods;

**Sets:** Creating set, set operations, Set Quantification with all and any;

**Files:** Text files, writing a text file, reading from a text file, writing a binary file, reading from a binary file, seek(), Structured text files, Binary data Interchange formats: CSV, HTML, XML, JSON, variables, Serialization using pickle module, Directories, Pickling;

**Exceptions:** Errors and exceptions, handling exceptions, modules. [13 hours]

### Unit IV

**Regular Expressions:** Defining regular expressions, Compiling regular expressions, using regular expressions, Using match objects to extract a value, Extracting multiple items, Replacing multiple items;

**DB_API, Graphics:** Python's standard web libraries-HTTPLib, URLLib, SMTPLib, The simplest Python Web server, Web Server Gateway Interface (WSGI), GUIs, Plots, Graphs, Visualization, Database connectivity [13 hours]

### Text Books:

1. Richard L. Holterman," Fundamentals of Python Programming", 2017

### Reference Books:

1. Allen Downey, Jeffrey Elkner, Chris Meyers, "How to Think Like a Scientist –Learning with Python ", First Edition, Green Tea Press, 2002,
2. Charles Dierbach," Introduction to Computer Science Using Python", Wiley Publication
3. Learning with Python ", Green Tea Press, 2002, First Edition.
4. Dave Kuhlman," A Python Book: Beginning Python, Advanced Python, and Python Exercises".

5. Mark Lutz, “Learning Python”,5th Edition”, O’Reilly
6. Micheal Urban, Joel Murach, “Murachs Python Programming”, 2016
7. Zed Shaw,” Learn Python the Hard way”, 3rd Edition, Addison Wesley,2013
8. Bill Lubanovic,” Introducing Python- Modern Computing in Simple Packages”, O ‘Reilly Publications

## CA14L: COMPUTER NETWORKS LAB

**Course Outcomes:**
**COP14.1:** Gain knowledge on using simulation tools.
**COP14.2:** Implement concepts of socket connections, file transfer process and congestion control algorithms.

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | 25 | 25 | 50 |

### PART A

1. Execute the following commands: arp, ipconfig, hostname, netdiag, netstat, nslookup, pathping, ping route, tracert.

2. Practically implement the cross-wired cable and straight wired cable using crimping tool.
3. Study of network IP address configuration: (Classification of address, static and dynamic address)
4. Study of network devices: (Switch, Router, Bridge)
5. Configure and Connect the computer in LAN.
6. Block the website using "Windows Defender Firewall" in windows 10. Share the folder in a system and access the files of that folder from other system using IP address.
7. Share the printer in Network and take print from other PC.
8. Configuration of Wi-Fi hotspot and connect other devices (mobile / laptop).
9. Configuration of VLAN using Packet Tracer/ GNS3.
10. Configuration of VPN using Packet Tracer/ GNS3.

### PART B

11. Write a socket program for implementation of echo.
12. Write a client-server application program for chat using TCP.
13. Write a program to perform File Transfer in Client & Server Using TCP/IP.
14. Write a program to implement Remote Command Execution (RCE).
15. Write a program to implement simple client-server application using UDP.

### Scheme of Evaluation:

| | |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05 Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |

# CA15L: PYTHON PROGRAMMING LAB

**Course Outcomes:**

**COP15.1:** Gain programming skills in Python to solve simple problems using looping structures, decision structures, functions, modules strings, lists, tuples, dictionaries, sets.

**COP15.2:** Implement concepts of objects, inheritance, exception handling, file handling, data visualization using Python programming.

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | 25 | 25 | 50 |

## Part A

1. Write Python program to Solve Quadratic Equation.
2. Write Python program to find Sum of Natural Numbers Using Recursion.
3. Write Python program to sort alphabetically the words from a string provided by the user.
4. Write Python program to perform the following operations on lists.
(i) Extend (ii) count (iii) sort (iv) reverse.
5. Write Python program to perform various set operations.
6. Write Python program to illustrate exception handling.
7. Write Python program to find the specific pattern in the input using regular expression.
Write a Python program to append text to a file and display the text.
8. Write Python program to add two numbers by taking input from the User.
9. Write Python program to draw a histogram to show the distribution of random values in a set of data.
10. Write Python program to create a csv file, read and print the contents.

## Part B

Any 6 Application based programs

### Scheme of Evaluation:

| | |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05 Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |

# IV SEMESTER

# CA16T: SOFTWARE ENGINEERING

**Objective:** To impart knowledge on Software Products and Process, various models, Software Development phases, Software reliability, Software verification and validation

**Skills to be developed:** to understand a real-life problem, analyse, and design software.

**Course Outcomes:**

**CO12.1:** Gain knowledge on software process models, professional responsibility, computer-based system engineering, requirements and specifications.
**CO12.2:** Understand software prototyping, user interface prototyping, and software design and domain specific architecture.
**CO12.3:** Gain knowledge on object oriented and function-oriented design, user interface design.
**CO12.4:** Gain knowledge on software reliability metrics, statistical testing, fault avoidance and tolerance, exception handling, defensive programming, software reusability.
**CO12.5:** Gain knowledge on software testing and its importance, test planning and strategies, project management, quality management, cost estimation and software maintenance.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | **3** | **4** | **40** | **60** | **100** |

## Unit - I

**Introduction:** Software Products and Software process, Process models: Waterfall modal, Evolutionary Development, Bohemia's Spiral model, Overview of risk management, Process Visibility, Professional responsibility. Computer based System Engineering: Systems and their environment, System Procurement, System Engineering Process, System architecture modelling. Human Factors, System reliability Engineering. Requirements and Specification: The requirement Engineering Process, The Software requirement document, Validation of Evolution of requirements, Viewpoint – oriented & method-based analysis, system context, Social 7 organizational factors. Data flow, Semantic, Objects, models, Requirement Specification, Non-functional requirement. [13 Hours]

## Unit - II

**Software Prototyping:** Prototyping in software process, Prototyping techniques, User interface prototyping. Software Design: Design Process, Design Strategies, Design Quality, System Structuring control models, Modular decomposition, Domain Specific architecture.
[13 Hours]

## Unit - III

**Object Oriented & function-oriented design:** An overview of object design, object Classes and inheritance Object identification, An object-oriented design example, Concurrent Objects, Data flow design Structural decomposition, Detailed Design, A Comparison of design Strategies. User interface design: Design Principles, User System interaction, Information Presentation, User Guidance, Interface Evaluation. [13 Hours]

## Unit - IV

**Software Reliability and reusability:** Software reliability metrics, Software reliability Specification, Statistical testing, Fault avoidance & tolerance, Software development with reuse, managing reuse, Generator based reuse, Application System Portability.
**Software Verification and Validation:** The testing Process, Test Planning & Strategies, Testing types, Structural and interface testing, Program inspections, mathematically based verification, Static analysis tools, clean room software development. Management Issues: Project management, Quality management, Software cost estimation, Software maintenance.
[ 13 Hours]

## Text Books:

1. Ian Sommerville**, "**Software Engineering", 9th Edition, Pearson Education Ltd. Publications, 2010.

**Reference Books:**

1. Roger S. Pearson, "Software Engineering, A Practitioner's approach", 7th Edition, Mc Graw Hill Publication, 2010.
2. Bernd Brugge, Alan H Dutoit, "Object -Oriented software Engineering", 3rd Edition, Pearson Education

# CA17T: DESIGN AND ANALYSIS OF ALGORITHM

**Objective:** to impart knowledge on design and analysis of algorithms for real life problems

**Skills to be developed:** To apply important algorithmic design paradigms and methods of analysis for computer solution of real-life problems.

**Course Outcomes:**

**CO17.1:** Gain knowledge on algorithms, Design and Analysis Framework, Asymptotic notations, Analysis of Non-recursive and Recursive algorithms and various algorithm design paradigms.

**CO17.2:** Ability to derive and solve problems using brute force, time complexities of algorithms using divide and conquer, decrease and conquer, transform and conquer strategies.

**CO17.3:** Ability to solve problems space and time tradeoffs, dynamic programming and Greedy method,

**CO17.4:** Solution for problems on Backtracking and Branch and Bound techniques, finding a feasible solution for decision problems, P and NP problems

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| 60 | 3 | 4 | 40 | 60 | 100 |

**UNIT - I** **[13 Hours]**

**Introduction:** Algorithms, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental Data Structures. **Fundamentals of the Analysis of Algorithm Efficiency:** The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms, Empirical Analysis of Algorithms

**UNIT – II** **[13 Hours]**

**Brute Force Method:** Selection Sort and Bubble Sort, Sequential Search, Brute-Force String Matching, Exhaustive Search, Depth-First Search and Breadth-First Search. **Decrease and Conquer:** Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decreaseby-a-Constant-Factor Algorithms. **Divide and Conquer:** Merge Sort, Quick Sort, Binary Tree Traversals and Related Properties, Strassen's Matrix Multiplication.

**UNIT – III** **[13 Hours]**

**Space and Time Tradeoffs:** Sorting by Counting, Input Enhancement in String Matching, Hashing. **Dynamic programming:** Binomial Coefficient, Principle of Optimality, Optimal Binary Search Trees, Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. **Greedy Technique:** Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees.

### UNIT – IV [13 Hours]

**Limitations of Algorithm Power:** Lower-Bound Arguments, Decision Trees, P, NP and NP Complete Problems. **Coping with the Limitations of Algorithm Power:** Back Tracking: n Queens problem, Hamiltonian Circuit Problem, Subset-Sum Problem. **Branch-and-Bound:** Assignment Problem, Knapsack Problem, Traveling Salesman Problem.

### Textbooks:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", 3rd Edition, Pearson, 2012.
2. Horowitz, Sahni, Rajasekaran, "Fundamentals of Computer Algorithms", 2/e, Universities Press, 2007.

### Reference Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, The MIT Press, 2009.
2. A.V. Aho, J.E. Hopcroft, J.D. Ullmann, "The design and analysis of Computer Algorithms", Addison Wesley Boston, 1983. 3. Jon Kleinberg, Eva Tardos, "Algorithm Design", Pearson Education, 2006.

## CA-C18T: INTERNET TECHNOLOGIES

**Objective:** To introduce Fundamentals of Web, Markup languages, scripting, Web designing
**Skills to be developed:** ability to develop web application.
**Course Outcomes:**
**CO18.1:** Understand the basics of Internet, web, web pages and www.
**CO18.2:** Usage of Forms, Frames in HTML, CSS style sheets and creation of web pages and web IR.
**CO18.3:** Ability to provide an insight on Javascript.
**CO18.4:** Understand basics of web development and to gain knowledge onweb and NoSQL databases.

| Total Number Of Hours | Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|---|
| 52 | 3 | 4 | 40 | 60 | 100 |

### UNIT-I

**INTERconnected NETwork:**

**Internet:** The Giant Wide Area Network, Communicating over the Internet. Accessing the Internet, Internet Organisations. Cyber Ethics. Internet Applications: Internet services, Electronic Mail(E-Mail), File Transfer, Real-Time User Communication, Remote Login, Usenet, **World Wide Web:** The Web, The Working Web, Web Terminology, Web Architecture, World Wide Web Challenges. [13 Hours]

### UNIT-II

**Hypertext Transfer Protocol (HTTP):** HTTP, HTTP Version, HTTP connections, HTTP Communication, Hypertext Transfer Protocol Secure, **Hypertext Transfer Protocol State Retention:** Cookies, Hypertext Transfer Protocol Cache, **Evolution of Web:** The Generations of Web, Web 1.0, Web 2.0, Web 3.0, **Big Data:** A Special Discussion, **Web IR:** Information Retrieval on the Web: Web Information Retrieval, Web Information Retrieval Tools, Web Information Retrieval Architecture (Search Engine Architecture). Web Information Retrieval Performance Metrics, Web Information Retrieval Models, Google PageRank [13 Hours]

### UNIT-III

**Web Development Basics:** Elements of Web Development, Client-Side and Server-Side Scripting, Model-View-Controller Architecture for Web Application Development, **Client-Side Technologies:** HTML. Hypertext Markup Language, CSS Cascading Style Sheets, JavaScript, Bootstrap Framework, AngularJS Framework, **Server-Side Technologies:** Server-Side Scripting, Personal Home Pages, **Node.js:** Server-Side JavaScript. [13 Hours]

### UNIT-IV

**Web Application Frameworks:** Django Ruby on Rails. **Web Databases:** Web Database, Structured Query Language: Relational Databases, **NoSQL Databases:** Non-relational and Distributed Data, Understanding Popular Databases. Research Trends on the Web: Contextual Information Retrieval, Web Mining [13 Hours]

### TEXT BOOKS:

1. Akshi Kumar, "Web Technology: Theory and Practice", CRC Press, 2019.
2. Robert W Sebesta, "Programming the World Wide Web", 4th Edition, Pearson Education, 2008.
3. Paul Deitel, Harvey Deitel, Abbey Deitel "Internet and World Wide Web: How to Program", Fifth Edition, Pearson, 2018

### REFERENCE BOOKS:

1. Web Technology: A Developer's Perspective, N.P. Gopalan and J. Akilandeswari, PHI, Learning, Delhi, 2013
2. Internetworking Technologies, An Engineering Perspective, Rahul Banerjee, PHI Learning, Delhi, 2011
3. M.Deitel, P.J.Deitel, A.B.Goldberg, "Internet & World Wide Web How to program", 3rd Edition, Pearson Education / PHI, 2004.
4. Chris Bates, "Web Programming Building Internet Applications", 3rd Edition, Wiley India, 2006.
5. Xue Bai et al, "The Web Warrior Guide to Web Programming", Thomson, 2003.
6. Sklar, "The Web Warrior Guide to Web Design Technologies", 1st Edition, Cengage Learning India.

## CA-C19L: ALGORITHMS LAB

**Course Outcomes:**

**COP.19.1:** Gain knowledge on fundamentals of Artificial Intelligence (AI) and expert systems, and knowledge representation.

**COP.19.2:** Application of basic principles of AI in solutions that require problem solving, knowledge representation.

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | **25** | **25** | **50** |

### PART A

1. Write a program to implement linear search algorithm Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.
2. Write a program to implement binary search algorithm. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.
3. Write a program to solve towers of honai problem and execute it for different number of disks
4. Write a Program to Sort a given set of numbers using selection sort algorithm. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
5. Write a program to find the value of an (where a and n are integers) using both brute-force based algorithm and divide and conquer based algorithm
6. Write a Program to Sort a given set of elements using quick sort algorithm. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.
7. Write a Program to find the binomial co-efficient C(n, k), [where n and k are integers and n > k] using brute force based algorithm and also dynamic programming based algorithm
8. Write a Program to implement Floyd's algorithm and find the lengths of the shortest paths from every pairs of vertices in a given weighted graph

### PART B

9. Write a program to evaluate a polynomial using brute-force based algorithm and using Horner's rule and compare their performances
10. Write a Program to solve the string matching problem using Boyer-Moore approach.
11. Write a Program to solve the string matching problem using KMP algorithm
12. Write a program to implement BFS traversal algorithm
13. Write a program to find the minimum spanning tree of a given graph using Prim's algorithm
14. Write a Program to obtain the topological ordering of vertices in a given digraph. Compute the transitive closure of a given directed graph using Warshall's algorithm.
15. Write a Program to Find a subset of a given set S = {s1,s2, ,sn} of n positive integers whose ..... sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1,2,6} and {1,8}. A suitable message is to be displayed if the given problem instance doesn't have a solution.

## CA-C20L: INTERNET TECHNOLOGIES LAB

**Course Outcomes:**

**COP8.1:** Creation of HTML form with various elements and writing JavaScript code for various operations, creating dynamic effects and including layers and basic animation and event handling.

**COP8.2:** Ability to apply the knowledge gained in creating small websites using HTML, CSS, JavaScript and XML.

| Credits | Hours/Week | Internal Assessment Marks | Exam | Total Marks |
|---|---|---|---|---|
| 2 | 4 | **25** | **25** | **50** |

### PART – A

1. Demonstrate E-Mail working (Sending, Receiving, forward)
2. How to create, organize meeting in Zoom/ GoogleMeet

3. Create a form by using various attributes of the input tags (text box, multiline textbox, option button, check box)
4. Create a simple HTML page by using some of the basic tags (hyperlink, marquee, image)
5. Create a web page with multiple types of style sheet used in a single page
6. Write a CGI sample program to send output back to the user
7. Create Time-Table using table tag
8. Creation of Frames in browser window using HTML.
9. Write a java script program to create dialogue boxes using alert, confirm and prompt methods
10. Write a java script program on Form Validations.
11. Write a java script program to perform four arithmetic operations: Addition, Subtraction, Multiplication and Division on two numbers.
12. Create a web site of our College.

## PART – B

Any 6 Web Based Application Programs

### Scheme of Evaluation:

| | |
|---|---|
| Writing two programs | - 10 Marks |
| Execution of one program | - 05 Marks |
| Viva | - 05 Marks |
| Record | - 05 Marks |
| **Total** | **- 25 Marks** |