

| SCSA1104 | PROBLEM SOLVING TECHNIQUES WITH<br>C AND C++ | L | T | P | Credits | Total Marks |
|----------|----------------------------------------------|---|---|---|---------|-------------|
|          |                                              | 3 | 0 | 0 | 3       | 100         |

**COURSE OBJECTIVES**

- To learn the fundamentals of PST and methodologies which are essential for building good C/C++ programs.
- To demonstrate a thorough understanding of modular programming by designing programs which require the use of programmer-defined functions.
- To impart the knowledge about pointers which is the backbone of effective memory handling.
- To demonstrate adeptness of object-oriented programming in developing solutions to problems demonstrating usage of data abstraction, encapsulation, and inheritance.

**UNIT 1 INTRODUCTION TO COMPUTER PROBLEM SOLVING**

9 Hrs.

Algorithms - Building blocks of algorithms (statements, control flow, functions) -Notation (pseudo code, flow chart) -Algorithmic problem solving for socio economic conditions in global perspectives - Simple strategies for developing algorithms (iteration, recursion) - Efficiency of algorithms.

**UNIT 2 BASICS OF C PROGRAMMING**

9 Hrs.

Introduction to C: Features of C - Structure of C Program-Data Types- 'C' Tokens-Input/output statements-Control Statement, Functions: – Types of Functions –Recursion

Algorithms: Reversing the digits of a number - Generation of Fibonacci sequence- Factorial Computation.

**UNIT 3 ARRAYS, STRINGS AND STRUCTURES**

9 Hrs.

Arrays: Single and Multidimensional Arrays— Array as Function Arguments, Strings: String Handling Functions, Structure: Nested Structures – Array of Structures – Structure as Function Argument–Function that Returns Structure, Union.

Algorithms: Sum of array elements- Removal of duplicates from an array-Finding the Kth smallest element.

**UNIT 4 POINTERS AND FILE PROCESSING**

9 Hrs.

Pointers: Introduction, Arrays Using Pointers – Structures Using Pointers – Functions Using Pointer, Dynamic Memory Allocation, Storage Classes, File Handling in 'C'.

Algorithms: Swap elements using Call by Reference – Sorting Arrays using pointers-Finding sum of array elements using Dynamic Memory Allocation.

**UNIT 5 OBJECT ORIENTED PROGRAMMING CONCEPTS**

9 Hrs.

Introduction-Procedure vs. object-oriented programming-Concepts: Classes and Objects-Operator & Function Overloading-Inheritance-Polymorphism and Virtual Functions.

Max.45 Hrs.

**COURSE OUTCOMES**

On completion of the course the student will be able to

- CO1: Develop solutions for the given problem.
- CO2: Design and Implement applications using arrays and strings.
- CO3: Develop and Implement applications using memory allocation and File concepts.
- CO4: Use proper class protection to provide security.
- CO5: Describe the reusability of code through Inheritance.
- CO6: Demonstrate the use of virtual functions to implement polymorphism.

**TEXT / REFERENCE BOOKS**

1. Dromey.R. G, "How to Solve it by Computer", Prentice-Hall of India, 8th Indian Reprint, 2008.
2. Aho.A.V., Hopcroft.J. E and Ullman.J.D. D, "The Design and Analysis of Computer Algorithms", Pearson education, 2004.Deitel and Deitel, C how to Program, 7th Edition, Pearson Education, 2013.
3. Brian W. Kernighan and Dennis M. Ritchie, "The C Programming Language", Pearson Education, 2015.
4. Bhave, Object Oriented Programming with C++, Pearson Education, 2004.
5. John R Hubbard, "Programming with C++", Schaums Outline Series, McGraw Hill, 2nd edition, 2009.
6. Bjarne Stroustrup, Programming: Principles and Practice using C++, 1st Edition, Addison Wesley Publications, 2008.

**END SEMESTER EXAM QUESTION PAPER PATTERN**

Max. Marks: 100

PART A: 10 Questions of 2 marks each - No choice

PART B :2 Questions from each unit with internal choice, each carrying 16 marks

Exam Duration: 3 Hrs.

20 Marks

80 Marks

| SCSA1204 | PYTHON PROGRAMMING | L | T | P | Credits | Total Marks |
|----------|--------------------|---|---|---|---------|-------------|
|          |                    | 3 | 0 | 0 | 3       | 100         |

**COURSE OBJECTIVES**

- To understand why Python is a useful scripting language for developers.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To build and package Python modules for reusability.
- To understand how to read and write files in Python.
- To learn how to use exception handling in Python applications for error handling.
- To design and program Python applications.

**UNIT 1 INTRODUCTION****9Hrs.**

History of Python- Introduction to the IDLE interpreter (shell) - Data Types - Built-in function - Conditional statements - Iterative statements- Input/output functions - Compound Data Types - Nested compound statements – Introduction to Object Oriented Concepts.

**UNIT 2 FILES AND EXCEPTIONS HANDLING, MODULES, PACKAGES****9Hrs.**

File Operations –Iterators - Exception handling - Regular Expressions- Creating Modules-Import Statement- Introduction to PIP-Installing Packages via PIP-Using Python Packages.

**UNIT 3 GUI PROGRAMMING****9 Hrs.**

GUI Programming in Python - Introduction to GUI library - Layout management - Events and bindings - Fonts – Colours - Canvas - Widgets (frame, label, button, check box, entry, listbox, message, radiobutton, text, spinbox).

**UNIT 4 DATABASE AND NETWORK****9Hrs.**

Database (using NoSQL): Connector Module –Cursor – Statements - Exceptions in database. Network connectivity: Socket module - Client – Server – Email – URLAccess.

**UNIT 5 CASE STUDY****9Hrs.**

Web Programming using Python Image Processing – Facebook Analysis – Twitter Analysis.

**Max. 45 Hrs.****COURSE OUTCOMES**

- CO1:** Describe the Numbers, Math functions, Strings, List, Tuples and Dictionaries in Python.
- CO2:** Do the decision making and write functions in Python.
- CO3:** Explain how to design GUI Applications in Python and evaluate different database operations.
- CO4:** Design and develop Client Server network applications using Python.
- CO5:** Ability to design real life situational problems and think creatively about solutions of them.
- CO6:** Apply the best features of mathematics, engineering and natural sciences to program real life problems.

**TEXT /REFERENCE BOOKS**

- Y. Daniel Liang, "Introduction to Programming Using Python", Pearson, 2013.
- Python Notes for Professionals by Stack Overflow Documentation (<https://books.goalkicker.com/PythonBook/>)
- Dr. Charles R. Severance, "Python for Everybody- Exploring Data Using Python 3", 2016.
- Paul Gries, Jennifer Campbell, Jason Montojo, "Practical Programming: An Introduction to Computer Science using Python 3", Pragmatic Bookshelf, 2<sup>nd</sup> Edition, 2014.
- Magnus Lie Hetland, "Beginning Python: From Novice to Professional", Apress.

**END SEMESTER EXAM QUESTION PAPER PATTERN****Max. Marks : 100****Exam Duration : 3 Hrs.****PART A :** 10 Questions of 2 marks each-No choice**20 Marks****PART B :** 2 Questions from each unit with internal choice, each carrying 16 marks**80 Marks**

| SCSA1205 | DATA STRUCTURES AND ALGORITHMS | L | T | P | Credits | Total Marks |
|----------|--------------------------------|---|---|---|---------|-------------|
|          |                                | 3 | * | 0 | 3       | 100         |

## COURSE OBJECTIVES

- To impart knowledge on the basic concepts of data structures and algorithms.
- To be familiar with writing recursive methods.
- To implement operations on Linked List, Stack and Queues.
- To implement traversal operations of trees and graphs.
- To understand concepts about various algorithm design techniques, searching and sorting techniques.

### UNIT 1 INTRODUCTION TO ALGORITHMS

9 Hrs.

Introduction to Data vs Information - Data Structures - Classification – Abstraction - Abstract data types (ADT) - Array - characteristics - Storage Representations. Array Order Reversal- Recursion- Array operations , Algorithm- complexity – Time and Space trade off.

### UNIT 2 LINKED LIST

9 Hrs.

Array Vs Linked List – Singly linked list - Representation of a linked list in memory - Operations on a singly linked list - Merging two singly linked lists into one list - Reversing a singly linked list – Polynomial Manipulation using List - Advantages and disadvantages of singly linked list - Circular linked list - Doubly linked list - Circular Doubly Linked List.

### UNIT 3 STACKS AND QUEUES

9 Hrs.

Introduction – Array Representation of a Stack – Linked List Representation of a Stack - Stack Operations - Algorithm for Stack Operations - Stack Applications: Tower of Hanoi - Infix to postfix Transformation - Evaluating Arithmetic Expressions. Queue – Introduction – Array Representation of Queue – Linked List Representation of Queue - Queue Operations - Algorithm for Queue Operations - . Queue Applications: Priority Queue.

### UNIT 4 TREES AND GRAPHS

9 Hrs.

Preliminaries of Tree ADT - Binary Trees - The Search Tree ADT–Binary Search Trees - AVL Trees - Tree Traversals - B-Trees - Heap Tree – Preliminaries of Graph ADT - Representation of Graph – Graph Traversal - BFS – DFS – Applications of Graph – Shortest - Path Algorithms – Dijkstra's Algorithm Minimum Spanning Tree – Prims Algorithm

### UNIT 5 ALGORITHM DESIGN TECHNIQUES, SEARCHING AND SORTING TECHNIQUES

9 Hrs.

Divide and Conquer Strategy – Greedy Algorithm – Dynamic Programming – Backtracking Strategy - List Searches using Linear Search - Binary Search - Fibonacci Search - Sorting Techniques - Insertion sort - Heap sort - Bubble sort - Quick sort - Merge sort - Analysis of sorting techniques.

Max.45 Hrs.

## COURSE OUTCOMES

On completion of the course the student will be able to

- CO1: Understand the concept of recursive algorithms.
- CO2: Demonstrate the different types of data structures.
- CO3: Able to understand the operations on linear data structures.
- CO4: Summarize searching and sorting techniques.
- CO5: Choose appropriate data structure as applied to specified problem definition.
- CO6: Understand and implement the various algorithm design techniques.

**TEXT /REFERENCE BOOKS**

1. Jean-Paul Tremblay, Paul G. Sorenson, 'An Introduction to Data Structures with Application', TMH, 2017.
2. Richard F. Gilberg, Forouzan, "Data Structures", Cengage, 2004, 2nd Edition.
3. Larry R. Nyhoff, ADTs, Data Structures, and Problem Solving with C++, Prentice Hall Edition, 2004.
4. Thomas H. Cormen, Charles E. Leiserson, "Introduction to Algorithms", 3rd Edition, 2010.

**END SEMESTER EXAM QUESTION PAPER PATTERN****Max. Marks : 100****Exam Duration : 3 Hrs.****PART A : 10 Questions of 2 marks each-No choice****20 Marks****PART B : 2 Questions from each unit with internal choice, each carrying 16 marks****80 Marks**

| SCSA2105 | PROBLEM SOLVING TECHNIQUES LAB | L | T | P | Credits | Total Marks |
|----------|--------------------------------|---|---|---|---------|-------------|
|          |                                | 0 | 0 | 4 | 2       | 100         |

### COURSE OBJECTIVES

- Identify the problem.
- To analyse the various steps in program development.
- Evaluate and select the best algorithm to solve the problem.
- Deploy suitable methods to get the desired output.
- Create the solutions for various Real-World Problems

### SUGGESTED LIST OF EXPERIMENTS:

1. Program to find GCD.
2. Program to find the max and min from the three numbers.
3. Program to find Exponentiation.
4. Program to find sum of an array of numbers.
5. Program to implement Sine function computation.
6. Program to Generate the Fibonacci sequence.
7. Program to find the roots of a Quadratic equation.
8. Program for reversing the digits of an integer.
9. Program to find the smallest divisor of an integer.
10. Program to Generate Prime Numbers.
11. Program to Raise a Number to a Large Power.
12. Program for Removal of Duplicates.
13. Program to find the kth smallest Element.
14. Program to generate histogram.
15. Program for addition and multiplication of matrices.
16. Program that converts a number ranging from 1 to 50 to Roman equivalent
17. To delete n Characters from a given position in a given string.
18. Program to search for a Key value in a given list of integers using linear search method.
19. Program to sort the number in ascending and descending order.
20. Program for finding the factorial using recursive and non-recursive functions

### COURSE OUTCOMES

On completion of the course, student will be able to

- CO1:** Analyse and classify the given problem into various modules
- CO2:** Analysing the complexity of problems, modularize the problems into small modules and then convert them into programs.
- CO3:** Develop the codes containing looping and decision-making statements.
- CO4:** Implement user defined functions.
- CO5:** Apply recursion and call the function with appropriate parameters.
- CO6:** Design and develop solutions to real world problems

| SCSA2203 | DATA STRUCTURES AND ALGORITHMS<br>LAB | L | T | P | Credits | Total Marks |
|----------|---------------------------------------|---|---|---|---------|-------------|
|          |                                       | 0 | 0 | 4 | 2       | 100         |

### COURSE OBJECTIVES

- To implement linear and non-linear data structures.
- To understand the different operations of search trees.
- To implement graph traversal algorithms and sorting algorithms.
- To get familiarized to binary tree traversal.
- To implement travelling salesman problem.

### LIST OF EXPERIMENTS

1. Program to find the largest and smallest number in an unsorted array.
2. Program to implement operations on a Singly linked list.
3. Program to implement operations on a doubly linked list.
4. Program to sort the elements using insertion sort.
5. Program to sort the elements using quick sort.
6. Program to sort the elements using merge sort.
7. Program to implement a Stack using an array and Linked list.
8. Program to implement Queue using an array and Linked list.
9. Program to implement Circular Queue.
10. Program to convert an infix expression to postfix expression.
11. Program to implement BFS and DFS
12. Program to implement N Queens problem.
13. Program to implement Binary Tree Traversal
14. Program to implement Travelling Salesman Problem

### COURSE OUTCOMES

On completion of the course, student will be able to

- CO1:** Remembering the concept of data structures through ADT including List, Stack and Queues
- CO2:** Understand basic concepts about stacks, queues, lists, trees and graphs
- CO3:** Able to apply and implement various tree traversal algorithms and ensure their correctness
- CO4:** Ability to analyze algorithms and develop algorithms through step by step approach in solving problems with the help of fundamental data structures.
- CO5:** Compare and contrast BFS and DFS.
- CO6:** Design applications and justify use of specific linear and binary data structures for various applications