SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
REVISIONS CARRIED OUT-2019-2020
2020-0DD

		L	T	P	Credits	Total Marks
SCSA1101	**PYTHON AND PROBLEM SOLVING TECHNIQUES**	**3**	**0**	**0**	**3**	**100**

COURSE OBJECTIVES

- To learn fundamentals of PST & basics of python programming.
- To learn modular programming.
- To study in detailed data representation, operator, control structures, functions in python and apply it on problems.
- To learn in built functions in python & apply it on mathematical problem.
- To learn advanced algorithm by implementing lists, tuples and dictionaries.

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), the efficiency of algorithms.

UNIT 2 DATA, EXPRESSIONS, STATEMENTS 9 Hrs.

Python interpreter and interactive mode; values and types: int, float, boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments; modules and functions, function definition and use, flow of execution, parameters and arguments; Algorithms: exchange the values of two variables, circulate the values of n variables, distance between two points.

UNIT 3 CONTROL FLOW, FUNCTIONS 9 Hrs.

Conditional statements: Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iterative statements: while, for, break, continue, pass, Applying for critical conditions. Algorithms: square root, exponentiation, sum an array of numbers, factorial computation-Sine function computation-Generation of the Fibonacci sequence-Reversing the digits of an integer.

UNIT 4 FACTORING METHOD 9 Hrs.

Functions: return values, parameters, local and global scope. Lambda, filters, map and reduce functions. Algorithms: Thesmallest divisor of an integer-The greatest common divisor of two integers-Generating Prime Numbers-Computing the Prime Factors of an integer- Raising a Number to a Large Power

UNIT 5 LISTS, TUPLES, DICTIONARIES 9 Hrs.

Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples: tuple assignment, tuple as return value; Dictionaries: operations and methods; advanced list processing – list comprehension; Algorithms: Removal of Duplicates -Partitioning -Finding the kth smallest Element- histogram.

Max. 45 Hrs.

COURSE OUTCOMES

On completion of the course, student will be able to
CO1 - Design a modular solution for a given problem.
CO2 - Choose appropriate algorithm for each module considering economic and social aspects.
CO3 - Employ multiple critical and creative thinking strategies in reasoning and problem solving.
CO4 - Evaluate and choose an appropriate problem solving strategy.

CO5 - Document the approach to be implemented in any language.
CO6 - Apply to the problem solving techniques on real world problems.

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, “The Design and Analysis of Computer Algorithms”, Pearson education, 2004.
3. Martin C. Brown, “ Python: The Complete Reference”, McGraw Hill, 2018.
4. John M. Zelle , “Python Programming: An Introduction to Computer Science”, Library of Congress.

END SEMESTER EXAMINATION 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**

SCSA1104	PROBLEM SOLVING TECHNIQUES WITH 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, “The Design and Analysis of Computer Algorithms”, Pearson education, 2004.Deitel and Deitel, C how to Program, 7th Eition, 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 **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**

2019-0DD

SCSA2101	PYTHON AND PROBLEM SOLVING TECHNIQUES LAB	L	T	P	Credits	Total Marks
		0	0	4	2	100

COURSE OBJECTIVES

- Identify the problem.
- List the possible ways to obtain the solution.
- Evaluate and Select the Best algorithm to solve the problem.
- Deploy suitable methods to get the desired output.
- Call the methods in order.

SUGGESTED LIST OF EXPERIMENTS

1. Program to exchange the values of two variables.
2. Program to circulate the values of n variables.
3. Program to find distance between two points.
4. Program to find square root.
5. Program to find GCD.
6. Program to find Exponentiation.
7. Program to find sum an array of numbers.
8. Program to find factorial.
9. Program to implement Sine function computation.
10. Program to Generate the Fibonacci sequence.
11. Program for Reversing the digits of an integer.
12. Program to find the smallest divisor of an integer.
13. Program to find the greatest common divisor of two integers.
14. Program to Generate Prime Numbers.
15. Program to Compute the Prime Factors of an integer.
16. Program to Raise a Number to a Large Power.
17. Program for Removal of Duplicates.
18. Program for Partitioning.
19. Program to find the kth smallest Element.
20. Program to generate histogram.

COURSE OUTCOMES

On completion of the course, student will be able to

CO1 - Analyze and classify the given problem into various modules.

CO2 - Design the program with basic syntax by reading input from the user and generating the desired output.

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.

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