COURSE CODE: 21ICS1   **NO. OF HOURS: 60**

Course Objective :

- Define and describe fundamental concepts of computers, algorithms, flowcharts and problem solving techniques.
- Illustrate basic knowledge of mathematical factoring methods to model an algorithm, flowchart and be able to solve given problem.
- Define merging, sorting, searching and text processing techniques to develop algorithms.
- Analyse a given problem, define an appropriate array technique and illustrate with an example

Learning Outcome :

- Examine the fundamental concepts of computers, algorithms, flowcharts and problem solving techniques.
- Explain basic knowledge of mathematical factoring methods to model an algorithm, flowchart and be able to solve given problem.
- Compare and interpret merging, sorting, searching and text processing techniques to develop algorithms.
- Construct a given problem, define an appropriate array technique and illustrate with an example .

UNIT - I   **15 Hours**
Introduction: Programs and algorithm, The Role of Algorithms in Computing, Algorithms as a technology, analyzing algorithms, Designing algorithms, Growth of Functions, Asymptotic notation, Standard notations and common functions.

Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of the Fibonacci sequence, Reversing the digits of an integer, Character to number conversion.

UNIT – II   **15 Hours**
C Programming: Getting Started, Variables and Arithmetic expressions. Input and Output: Standard input and output, formatted output- printf, variable length argument list, formatted input-scanf.

Control Flow: Statements and Blocks, If-else, else-if, switch, loops: while loop, for loop, do while, break and continue, goto and labels.

Functions and category of functions, Pointers, Pointers and Arrays: pointers and address, pointers and function arguments, arrays, multidimensional array, initialization of pointer arrays, command line arguments.

UNIT - III   **15 Hours**
Factoring Methods: Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, computing the prime factors of an integer, raising a number to a large power.

Array Techniques: Array order Reversal, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, Finding the kth smallest element, multiplication of two matrices.

UNIT - IV   **15 Hours**
Merging: the two-way merge, Sorting: Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning.

Searching: linear search, binary search, hash search. Text processing and Pattern searching: text line length adjustment, keyword searching in text, linear pattern search

Text Books:
1. R.G.Dromey, "How to Solve it by Computer", Pearson Education India, 2008.

2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction

to Algorithms", 3rd Edition, The MIT Press Cambridge, Massachusetts London, England,

2008

 3. Brain M. Kernighan, and Dennis M. Ritchie, "The C Programming Language", 2nd

edition, Princeton Hall Software Series, 2012.

Reference Books:
1. Steven S. Skiena, "The Algorithm Design Module", 2nd Edition, Springer-Verlag London Limited, 2008.
2. Donald E. Knuth, The Art of Computer Programming", Volume 1: Fundamental Algorithms, 3rd Edition, Addison Wesley Longman, 1997.
3. Donald E. Knuth, The Art of Computer Programming", Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.
4. Greg Perry and Dean Miller, "C programming Absolute Beginner's Guide", 3rd edition, Pearson Education, Inc, 2014.
Web Resources:
1. http://algorithmsforinterviews.com "Algorithms for Interviews"

PROBLEM SOLVING LAB USING C

Course Objective:

• Define and explain the syntax and construction of C program.

• Demonstrate writing, compiling and execution of the C program.

• Illustrate all the concepts that have been covered in the theory course.

• Evaluate the problem and understand the flow using design techniques.

Learning Outcome :

• Construct the syntax of C program.

- Ability to write, compile and execute C programs.

- Analysing all the concepts with proper examples.

- Asses the programming skills of each student through examples.

Write, and execute C program for the following:

1. To read radius of a circle and to find area and circumference

2. To read three numbers and find the biggest of three

3. To check whether the number is prime or not

4. To read a number, find the sum of the digits, reverse the number and check it for palindrome

5. To read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers

6. To read percentage of marks and to display appropriate message (Demonstration of else-if ladder

7. To find the roots of quadratic equation

8. To read marks scored by n students and find the average of marks (Demonstration of single dimensional array)

9. To remove Duplicate Element in a single dimensional Array

10. To perform addition and subtraction of Matrices

11. To find factorial of a number

12. To generate Fibonacci series

13. To remove Duplicate Element in a single dimensional Array using function

14. To find the length of a string without using built in function

15. To demonstrate string functions

16. To read, display and add two m x n matrices using functions

17. To read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.

18. To Swap Two Numbers using Pointers

19. To demonstrate student structure to read & display records of n students

20. To demonstrate the difference between structure & union.

OPEN ELECTIVE
FUNDAMENTALS OF COMPUTERS AND C PROGRAMMING LANGUAGE

Course Objective (Cos):
1. It will help in understanding the basics of computers.
2. Flowcharts will allow visualizing the problem and designing the solution for it.
3. It will develop problem solving skills.
4. It will create a foundation for programming.

UNIT - I: 05 HOURS

Introduction to Computers, Simple model of a computer, Characteristics of a computer, **Input-Output devices:** Keyboard, mouse, monitor, MICR, OMR, Bar code, **Printers:** Inkjet, laser, dot matrix, **Computer Memory:** Read Only Memory, Random Access Memory, Flash Memory, Magnetic Hard Disk.

UNIT - II: 05 HOURS

Algorithms**:** Definition, Characteristics of algorithms, Example of an algorithm. Problem solving using computers, **Flowchart:** Symbols used in flowcharts, standard conventions of flowchart, examples.

UNIT - III: 05 HOURS

Basic Structure of a C program, character set, keywords, Identifiers, Data types, Constants and Variables, Data type Declaration statement, Assigning Values to a variable, Operators, Expressions.

UNIT - IV: 10 HOURS

Input/Output Functions (scanf, printf), Branching statements, Looping statements, Jumping statements, If statement, If-Else statement, switch statement, While statement, Do-while statement, Nested For statement.

UNIT - V: 05 HOURS

Arrays, Function Prototypes and Categories, Structure implementation.

LAB LIST:

NO. OF PRACTICAL HOURS: 15

1. Write a C program to find sum and average of three numbers
2. Write a C program to find the roots of a quadratic equation.
3. Write a C program to find both the largest and smallest number in a list of integers.
4. Write a C program to perform addition of two matrices.
5. Write a C program to find the factorial of the given number.

REFERENCE BOOKS:
1. V Rajaraman and Niharika Adabala, Fundamentals of Computers, 6th Edition, PHI Learning Private Limited.
2. Kanetkar, Yashavant, Let Us C, 4th Edition, BPB Publications
3. Balagurusamy, E Programming in ANSI C 2nd Edition. Tata McGraw – Hill