

# 16CS102 COMPUTER PROGRAMMING

Hours Per Week :

L	T	P	C
3	1	2	5

## Course Description and Objectives:

This course is aimed at offering fundamental concepts of programming language to the students. It starts with the basics of C-programming and deals with the structure and various attributes required for writing a 'C' program. It also introduces various operators and control statements used in programming. Then it switches to functions and arrays. It goes on with strings, pointers, files and the user defined data types. As a first-level course in computer science, it forms the basis to understand usage of various attributes in writing a program.

## Course Outcomes:

Upon completion of the course, the student will be able to

- CO1: Understanding of how to write simple, but complete C programs.
- CO2: Identification of suitable data types for operands and design of expressions having right precedence.
- CO3: Application of decision making and iterative features of C Programming language effectively.
- CO4: Design and development of problem specific data structures and accessing methods to build large modular programs.
- CO5: Development of C programs that is understandable, debuggable, maintainable and more likely to work correctly in the first attempt.

## SKILLS:

- ✓ *Identify suitable data types for an application.*
- ✓ *Apply control statements for decision making problems.*
- ✓ *Use multidimension array for matrix application.*
- ✓ *Design a program to calculate average of a class.*
- ✓ *Analyze the difference between static and dynamic memory allocation.*

UNIT - 1 L-9, T-3

INTRODUCTION TO C PROGRAMMING : Structure of C program- comments, processor statement, function header statement, variable declaration statement and executable statement; C character set - constants, identifiers, operators, punctuations, keywords, modifiers, identifiers, variables, c scopes, basic data types, type qualifiers, storage classes, reading and writing characters and formatted I/O.

UNIT - 2 L-9, T-3

OPERATORS AND CONTROL STATEMENTS : Operators- assignment, arithmetic, relational, logical, bitwise, ternary, address, indirection, size of, dot, arrow and parentheses operators; Expressions precedence of operators and associative rules; Control statements- category of statements, selection, iteration, jump, label, expression and block.

UNIT - 3 L-9, T-3

FUNCTIONS AND ARRAYS : Function- declaration, prototype, definition, calling by value and call by address, standard library functions and recursive functions; Array- declaration, initialization, reading, writing, accessing and passing as a parameter to functions, 2D-arrays and multidimensional arrays.

UNIT - 4 L-9, T-3

STRINGS AND POINTERS : Strings- declaration, string library functions, array of strings and command line arguments; Pointers- declaration, initializing pointers, multiple indirection, relationship between arrays and pointers; Scaling up- array of arrays, array of pointers, pointer to a pointer, pointer to an array; pointer to functions and dynamic memory allocation functions.

UNIT - 5 L-9, T-3

STRUCTURES AND FILES : Structures - declaration, initialization and accessing, array of structures and passing structures to functions, structure pointers, arrays and structures within structures, unions, bit-fields, types and enumerations; Files - I/O and processing operations on text and binary files; Pre-processor directives.

## LABORATORY EXPERIMENTS

Course Outcomes:

Upon successful completion of this course, the student will be able to:

- write, compile and debug programs in C language.
- formulate problems and implement algorithms in C.
- choose programming components that efficiently solve computing problems in real-world.

LIST OF EXPERIMENTS

Total hours: 30

1. Compute the factors of a number.
2. Compute the average of 'n' numbers.
3. Find whether a number is palindrome or not.
4. Find whether a number is a power of 2 or not.
5. Compute the factorial of a number.
6. Implement any kind of operation (+,-,\*,/,%) using a switch case.

ACTIVITIES:

- *Implement matrix operations.*
- *Implement malloc and calloc functions.*
- *Copy the content of one file into the other.*
- *Implement string manipulations functions.*

7. Swap two values using call by value and call by reference.
8. Using structure of arrays.
9. Find the reversal of a number.
10. Find the frequency of each number in the array.
11. Which takes 0's and 1's as input and the array should consist of all 0's first and then 1's.
12. Copy the first 10 words of a file into the other file.
13. Count the number of words in a file.
14. Create a structure which stores the student's information in a class.
15. Reverse the contents of the array.
16. Implement pointer of pointers.
17. Give  $n^{\text{th}}$  term of the Fibonacci number.
18. Find the factorial of a number using recursion.
19. Find the number of vowels in a file.
20. Access the structure and union members.

TEXT BOOK:

1. A. Mittal, "Programming in C - A Practical Approach", Pearson Education, India, 2015

REFERENCE BOOKS:

1. R. Thareja, "Introduction to C Programming", 2<sup>nd</sup> edition, Oxford University Press India, 2015.
2. C. H. Schildt, "The Complete Reference", 4<sup>th</sup> edition, Tata McGraw-Hill, 2000.
3. E. Balagurusamy, "Programming in ANSI C", 4<sup>th</sup> edition, Tata McGraw-Hill, 2008.