

C - PROGRAMMING

Lectures	: 4 periods / week	Sessional Marks	: 40
Tutorials	: -- period / week	Semester End Exam Marks	: 60
Semester End Exam	: 3 hrs	Credits	: 4

Course Objectives:

- Be familiar with computer software and hardware components, how they interact and its block diagram.
- Understand the basic problem-solving process using algorithm, Flow Charts and pseudo-code development.
- Understand the phases of compilation, from preprocessing through linking and loading. Learn how to customize compilation to produce intermediate files, etc
- Be well-versed with various conditional and iterative structures and should be able to use them effectively for efficient programming.
- Able to recognize the need for arrays and develop thorough knowledge on the concept of numerical and character arrays and get a better handle on multi-dimensional arrays, pointers, Learn to effectively use pointers for Dynamic memory allocation.
- Learn to use structures and unions to create custom data types in C.
- Have basics in File Operations
- Have sound theoretical and practical knowledge in C .

Course Outcomes:

- Thorough understanding of basic components of a computer and their operations.
- The ability to be equipped with the basic problem-solving skills using algorithm, flow charts and pseudo-code.
- Thorough knowledge about various phases of compilation, from preprocessing through linking and loading. Learn how to customize compilation to produce intermediate files, etc.
- The ability to use the control structures effectively to write efficient programs.
- Sound knowledge regarding the numerical and character arrays
- Profound skills to develop various user-defined string handling functions which mimic the built-in string manipulation functions.
- Skills to control program's memory consumption by dynamically allocating and freeing memory as needed.
- The ability to use structures and unions and develop various user-defined data types in C.
- The basic knowledge to work with File I/O and perform various operations on sequential and random access files, including reading and writing text and binary data.
- Have sound theoretical and practical knowledge in C and could effectively use their skills to develop programs for complex applications.

UNIT – I

Introduction:

Computer Fundamentals: Computer & it's Components, Hardware / Software, Algorithm, Characteristics of algorithm, Flowchart, Symbols are used in flowchart, history of C, Basic structure of C, C language features.

C Tokens:

Character set, Variables, Keywords, Data types and sizes, Type qualifiers, Numeric Constants and their forms of representation, Character Constants, String Constants, Declarations and Initialization of variables.

Operators & Expressions:

Arithmetic operators, and expressions, Type-conversion rules, Coercion, Assignment operators and expressions, Increment and decrement operator, Conditional operator, Statements, Preprocessor directives, Input/ Output functions and other library functions. Relational operators and expressions. Boolean operators and expressions.

Programming Exercises for Unit I :

C-Expressions for algebraic expressions, Evaluation of arithmetic and boolean expressions. Syntactic errors in a given program, Output of a given program, Values of variables at the end of execution of a program fragment, Filling the blanks in a given program, Computation of values using scientific and Engineering formulae, Finding the largest of three given numbers.

UNIT – II**Conditional Statements:**

Blocks, If-Else statement, Else-If statement and Switch statement.

Iterative Statements:

While loop, For loop, Do-While loop, Break, and continue.

Arrays:

One - dimensional and character arrays, Two-dimensional numeric arrays.

Programming Exercises for Unit - II:

Computation of discount on different types of products with different ranges of discount Finding the type of triangle formed by the given sides, Computation of income-tax, Computation of Electricity bill, Conversion of lower case character to its upper case, Finding the class of an input character; Sum of the digits of a given number, Image of a given number, To find whether a given number is-prime; Fibonacci; abundant; perfect, Strong, Armstrong; deficient, Prime factors of a given number, Merging of lists, Transpose of a matrix, Product and sum of matrices, String processing-length of a string; comparison of strings; reversing a string; copying a string, Sorting of names using arrays, Graphics patterns, To print prime numbers and Fibonacci numbers in a given range, and Amicable numbers.

UNIT – III**Functions:**

Function Definition, types of User Defined Functions, Parameter passing mechanisms, and simple recursion.

Scope & extent:

Scope rules, Storage Classes, Multi-file compilation.

Pointers:

Pointers Arithmetic, Character array of pointers, Dynamic memory allocation, array of Pointer, Pointer to arrays.

Programming Exercises for Unit - III:

Recursive Functions: factorial, GCD(Greatest Common Divisor), Fibonacci; To evaluate the pointer arithmetic expressions; An interactive program to perform Pointers & Functions - Insertion sort, Bubble

sort, Linear search Binary search, Computation of Statistical parameters of a given list of numbers, Counting the number of characters, words and lines in a given text, Table of values of $f(x,y)$ varying x and y ; Using Storage Classes to implement the multifile compilation; implement the string operations using Dynamic memory allocation functions;

UNIT – IV

Structures:

Structures, Array of structures, structures within structures, Pointer to structures, self referential structures, Unions.

Files:

File Handling functions, File error handling functions, Command-line arguments.

Programming Exercises for Unit - IV:

Operations on complex numbers, operations on rational number (p/q form), Matrix operations with size of the matrix as a structure; Frequency count of keywords in an input program, Sorting a list of birth records on name and date of birth using File handling functions, Student marks processing, Library records processing - sorting on name, author, Copy one file to another.

TEXT BOOK:

1. Programming with C (Schaum's Outlines) by Byron Gottfried, Tata Mcgraw-Hill, 2010.
2. Programming with C by K R Venugopal & Sudeep R Prasad, TMH., 1997

REFERENCE BOOKS:

1. Programming in C by Pradip Dey and Manas Ghosh ,Second Edition,OXFORD
2. 'C' Programming by K.Balaguruswamy, BPB.
3. C Complete Reference, Herbert Sheildt, TMH., 2000

WEB REFERENCES:

1. <http://lectures-c.blogspot.com/>
2. http://www.coronadoenterprises.com/tutorials/c/c_intro.htm
3. <http://www.cprogramming.com/tutorial/c/lesson1.html>
4. http://vfu.bg/en/e-Learning/Computer-Basics--computer_basics2.pdf
5. <http://cprogramminglanguage.net/>