

SEMESTER-II
COURSE 4: PROGRAMMING IN C

Theory

Credits: 3

3 hrs/week

Course Objectives:

- Provides knowledge on Algorithms, Flow chart and basic programming language.
- Provides complete knowledge of C language.
- Helps to develop logics which will help them to create program and applications in C.
- Learning the basic programming constructs, they can easily switch over to any other language in future.

Course Outcomes:

Upon successful completion of this course, students will be able to-

- Understand the basic terminology used in computer programming.
- Write, compile and debug programs in C language.
- Use different data types in a computer program.
- Design programs involving decision structures, loops and functions.
- Understand the dynamics of memory by the use of pointers and Structures.
- Apply different operations in File handling.

Syllabus

UNIT - I:

Introduction to Algorithms and Programming Languages: Algorithm - Key features of Algorithms - examples of Algorithms, Flow Charts– Pseudo code, Programming Languages – Generation of Programming Languages – Structured Programming Language.

Introduction to C: Introduction – Structure of C Program, Writing the first C Program, File used in C Program – Compiling and Executing C Programs, Using Comments – Keywords – Identifiers, Basic Data Types in C, Variables – Constants, I/O Statements in C, Operators in C, Programming Examples, Type Conversion and Type Casting.

Case Study:

Enter any alphabet and display whether it is vowel or a consonant.

UNIT - II:

Control Structures and Functions: Decision Control and Looping Statements: Introduction to Decision Control Statements, Conditional Branching Statements, Iterative Statements, Nested Loops, Break and Continue Statement – Go to Statement. Functions: Introduction, Using functions – Function declaration/ prototype – Function definition, Function call – Return statement – Passing parameters, Scope of variables, Storage Classes, Recursive functions.

Case Study:

Print first 10 natural numbers. (using while, do-while, for loop , break, continue.)

UNIT - III:

Arrays: Introduction, Declaration of Arrays, accessing elements of the Array – Storing Values in Array, Calculating the length of the Array, Operations that can be performed on Array, Passing one dimensional array to function.

Two dimensional Arrays, accessing two dimensional arrays, Passing two dimensional arrays to functions.

Strings: Introduction, String Operations using String functions.

Case Study:

Searching an element in an array.

Disadvantages of an array.

UNIT - IV:

Pointers, Structures and Unions: Pointers: Understanding Computer Memory – Introduction to Pointers, Declaring Pointer Variable, Pointer Expressions and Pointer Arithmetic – Null Pointers, Passing Arguments to Functions using Pointer, Pointer and Arrays – Passing Array to Function, Memory Allocation in C Programs, Memory Usage – Dynamic Memory Allocation, Drawbacks of Pointers. Structures: Introduction to structures, Nested Structures. Union, and Enumerated Data Types: Introduction to Union – accessing union elements, Enumerated Data Types.

Case Study:

Difference between Arrays, structures & unions

UNIT – V:

File Handling: Files: Introduction to Files, Using Files in C, Reading Data from Files, Writing Data from Files, Detecting the End-of-file, Error Handling during File Operations.

Case Study:

Write a program to read a text file, convert all the lowercase characters into upper case and rewrite the uppercase characters in the file.

PRESCRIBED TEXT BOOKS:

Computer Fundamentals and Programming in C by REEMA THAREJA from OXFORD UNIVERSITY PRESS

REFERENCE BOOKS:

1. E Balagurusamy, COMPUTING FUNDAMENTALS & C PROGRAMMING – Tata McGraw-Hill, Second Reprint 2008, ISBN 978-0-07-066909-3.
2. Ashok N Kamthane: Programming with ANSI and Turbo C, Pearson Edition Publ, 2002.
3. Henry Mullish&HuubertL.Cooper: The Sprit of C, Jaico Pub, House,1996.
4. Teach your C Skills-Kanithker

SEMESTER-II
COURSE 4: PROGRAMMING IN C

Practical

Credits: 1

2 hrs/week

List of Experiments

1. Write a C program to calculate the expression: $((a*b)/c)+(a+b-c)$.
2. Write a C program to calculate $(a+b+c)^3$.
3. Write a C program to check whether the given number is Prime or Not.
4. Write a C program to find the sum of individual digits of a given number .
5. Program to convert Hours into seconds.
6. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
7. Write a program to check whether the given number is Palindrome or Not.
8. Write a C program to check whether a given 3-digit number is an Armstrong number or not.
9. Write a C program to print the numbers in triangular form.

```
1
1 2
1 2 3
1 2 3 4
```

10. Program to display the number of days in given month using Switch – Case.
11. Write a C program to perform the following:
 - I. Addition of two matrices.
 - II. Multiplication of two matrices.
12. Write a C program to determine if the given string is a palindrome or not.
13. Write C program to find the factorial of a given integer using a recursive function.
14. Write a C program to concatenate two strings using pointers.
15. Write a C program to find the length of a string using pointers.
16. Program to display Student Details using Structures.
17. Write a C program to
 - I. Write data into a File.
 - II. Read data from a File.