CLUSTER UNIVERSITY SRINAGAR

SYLLABUS (FYUP UNDER NEP 2020)

Offered By Department of COMPUTER APPLICATIONS

Semester 2nd (Major Course)

Course Title: Programming with C

Course Code: UGCOA22J201
Credits: 4 (Theory: 3, Practical: 1)
Contact Hrs: 75 (Theory: 45, Practical: 30)

Max. Marks 100
Theory External: 60; Min Marks: 24
Theory Internal (Continuous Assessment): 15 Marks, Min Marks: 06
Practical Experimental Basis= 15, Min. Marks: 06
Practical Experimental (Continuous assessment) = 10, Min. Marks: 04

Objectives

- To impart adequate knowledge on the need of programming languages and problem-solving techniques.
- To provide exposure to problem-solving through C programming.
- To familiarise the basic and advanced syntax and semantics of C Language.

UNIT 1: (15 Hrs)

Problem Solving: Introduction, Steps in Problem Solving, Approaches to problem solving, Flowcharts and Algorithms (Definition, Symbols & Characteristics), Simple Examples of Flowcharts and Algorithms (Real Life Examples), Overview of Problem Instances, Categories of Computer Languages, Concept of Debugging,

Types of Program errors. Introduction to C- Language: History, Features, Structure & Life Cycle of a C-Program. Data types and sizes, Variables, Constants, Keywords, Storage Classes, Operators (Unary, Arithmetic, logical, Bitwise, Assignment, Ternary), Expressions

UNIT 2: (15 Hrs)

Control statements (if- else, switch, break, continue, goto), Loops (for, while, do-while), Macros & Preprocessor Directives. Arrays, Functions & Sorting: Arrays (1D and Two-dimensional); String handling; Functions (built-in and user defined), declaration, definition, and function call, parameter passing and return types, Recursion. Sorting,

UNIT 3: (15 Hrs)

Introduction to Pointers, Pointer Arithmetic, Call by Value v/s call by Reference, Structure &, Union: Declaration, Accessing structure and union elements, difference, Array of structures, Nested structures, passing Arrays to functions, Basics of File Handling: Introduction to Stream, File Access Modes, Text v/s Binary Files, File I/O Operations, Command Line arguments.

Sample Lab Work (indicative) (1 Credits, 30 Hours)

1. Write a C program to declare and initialise variables of different data types.
2. Write a C program to perform arithmetic operations on variables.
3. Write a C program to calculate the sum of two numbers.
4. Write a C program to use different control flow statements (if-else, switch, break, continue, goto) to control the flow of a program.
5. Write a C program to compare two variables and print the result.
6. Write a C program to find the largest of three numbers.
7. Write a C program to use different looping statements (for, while, do-while) to iterate over a set of statements.
8. Write a C program to check if a number is prime.
9. Write a C program to find the prime numbers between two given numbers.
10. Write a C program to find the Fibonacci sequence up to a given number.
11. Write a C program to print the pattern:

```
*
* *
* * *
* * * *
* * * * *
```

9. Write a C program to define and use a macro.
10. Write a C program to use different preprocessor directives.
16. Write a C program to declare and initialise an array of integers.
17. Write a C program to access and modify the elements of an array.
18. Write a C program to find the sum, average, and maximum element of an array.
19. Write a C program to copy the elements of one array to another array.
20. Write a C program to reverse the order of the elements of an array.
21. Write a C program to reverse a string.
22. Write a C program to define and use a function to calculate the sum of two numbers.
23. Write a C program to define and use a function to find the largest of three numbers.
24. Write a C program to find the factorial of a number.
25. Write a C program to sort an array using any sort algorithm.
26. Write a C program to declare and define a structure to store the details of a student (name, roll number, marks) and access and modify its members.
27. Write a C program to pass a structure to a function.
28. Write a C program to declare and define a union to store different data types.
29. Write a C program to declare and initialise a pointer variable and access and modify the values pointed by a
pointer variable.
30. Write a C program to pass a pointer variable to a function.

SUGGESTED READING:

1. E. Balagurusamy, Programming in ANSI C, Tata McGraw Hills, New Delhi.
2. Brian W Kernighan and Dennis C. Ritchie, The C Programming Language, PHI Pvt. Ltd.
3. H. Schildt, A Complete Reference in C, Tata McGraw Hills, New Delhi.
4. Y. Kanetkar, Let Us C, BPB, New Delhi.
5. R.G. Dromey, How to Solve it by Computer, Pearson Education.