SEMESTER 2nd
MAJOR COURSE

CAP222J: COMPUTER APPLICATIONS (PROGRAMMING FUNDAMENTALS THROUGH C)

CREDITS: 4 + 2

COURSE OBJECTIVES:

1. *To demonstrate the use of flowcharts and algorithms for problem solving*
2. *To introduce the concepts of structured programming*
3. *To familiarize the student with the syntactic constructs of C*
4. *To enable to the students to translate algorithms into C programs*

THEORY (4 CREDITS)

UNIT – I **(15 Lectures)**

Programming Languages: History and Role of Programming Languages, Syntax and semantics, source code and object code, datatypes, variables, constants, declaration, Structured Data Types. Sequence Control: Implicit and Explicit. Sequence control between Statements. Subprogram Control: Simple call return and recursive subprogram. Language Paradigms: Simple Procedural Languages, Block Structured Programming Languages, Object Based Languages, Functional Languages, Logic Programming Languages.

Flowcharts, Flowchart Elements, Problem Solving Through Flowcharts.

Algorithms, Characteristics of an Algorithm, Algorithms for basic problems.

UNIT – II

Introduction to C Programming **(15 Lectures)**

History and overview of C, Basic structure of a C Program, Compilation, Execution and Debugging of programs in C.

Keywords, Identifiers and Datatypes. Variables and Constants. Comments. Console I/O using printf() and scanf(). Typecasting.

Operators – Arithmetic, Logical, Relational, Increment Decrement and Assignment Operators. Expressions. Operator Precedence.

Conditional Statements (If, If-Else, If-Else If, Nested If, Switch).

UNIT – III **(15 Lectures)**

Looping, Functions and Pointers.

Loops (while, do-while, for). Break and Continue. Nested Loops.

Functions: Declaring, Defining and Calling. Call by Value, Call by Reference. Function Arguments and Return Values.

Pointers: Declaring and Initializing. Accessing value of a pointer variable. Pointer Expressions. Pointer Increments and Scale Factors. Pointers and Arrays. Passing Pointers to Functions.

UNIT – IV

Arrays, Strings, Structures and Unions

Declaring, Initializing 1-D arrays and 2-D arrays. Accessing Elements of an Array, Memory Layout of Arrays. Passing Arrays to Functions, Command Line Arguments.

Character Arrays and String. Declaring and Initializing Strings, Reading and Writing Strings, String Handling Function (strlen, strcat, strcmp, strcpy).

Structures and Unions: Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of structures, Passing structures to functions.

Dynamic Memory Allocation using malloc and free.

TEXTBOOKS:

1. Terence Pratt, Programming Languages Design and Implementation (Pearson/Prentice Hall)
2. Balagurusamy, Programming in ANSI C, 8th Edition (McGraw Hill)

REFERENCES:

1. Kanetkar - Let us C (BPB Publications)
2. Ghezzi, Jazayeri - Programming Language Concepts (Wiley)
3. Srivastava – C in Depth (BPB Publications)
4. Beej's Guide to C Programming
5. Byron Gottfried - Schaum's Outline of Programming with C (McGraw-Hill)

LABORATORY COURSE (02 CREDITS):

1. WAP to print the sum of digits of an integer.
2. WAP to print the product of digits of an integer.
3. WAP to reverse a number.
4. WAP to compute the sum of the first n terms of the following series S = 1+1/2+1/3+1/4+……
5. WAP to compute the sum of the first n terms of the following series S =1-2+3-4+5…….
6. Write a program to check whether a given string is Palindrome or not. Convert this program into to a function that checks if a given string is a palindrome.
7. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
8. WAP to compute the factors of a given number.
9. Write a macro that swaps two numbers. WAP to use it.
10. WAP to print a triangle of stars as follows (take number of lines from user):

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

11. WAP to perform following actions on an array entered by the user:

i) Print the even-valued elements
ii) Print the odd-valued elements
iii) Calculate and print the sum and average of the elements of the array

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to create an array of user-defined size dynamically using malloc() function. Display the sum of values entered in it and use free() to release its memory.
16. Write a program to perform following operations on strings:
    a) Show address of each character in string
    b) Concatenate two strings without using strcat function.
    c) Concatenate two strings using strcat function.
    d) Compare two strings
    e) Reverse the string

***