BCA (HONOURS) 1st SEMESTER
DSC / CORE-1
BCA120C1: PROGRAMMING FUNDAMENTALS USING C/C++

CREDITS: THEORY: 4, PRACTICAL: 2

THEORY: 60 LECTURES (CREDITS: 4)

UNIT-I

1. **Introduction to C** **(5 Lectures)**
History of C, Overview of Programming, Using main() function, Compiling and Executing Simple Programs in C.

2. **Data Types, Variables, Constants, Operators and Basic I/O** **(5 Lectures)**
Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putcharetc), Formatted and Console I/O (printf(), scanf(), Using Basic Header Files (stdio.h, conio.hetc).

3. **Expressions, Conditional Statements and Iterative Statements** **(5 Lectures)**
Simple Expressions in C (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

UNIT-II

4. **Functions and Arrays** **(10 Lectures)**
Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.
Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Twodimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns)', Introduction to Multi-dimensional arrays

5. **Derived Data Types (Structures and Unions)** **(5 Lectures)**
Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

UNIT-III

6. **Pointers and References in C** **(9 Lectures)**
Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, using references as function arguments and function return values

7. **Memory Allocation in C** **(6 Lectures)**
Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation

UNIT-IV

8. File I/O, Preprocessor Directives **(5 Lectures)**
Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and writeQ functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifhdef and #undef), Macros

9. **Introduction to C++** **(10 Lectures)**
Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Class Variables &Functions, Objects as parameters, specifying the Protected and Private Access, Copy Constructors.

Reference Books
1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003 th
2. BjameStroustrup, "The C++ Programming Language", 4 Edition, Addison-Wesley, 2013.
3. BjameStroustroup, "Programming ~ Principles and Practice using C++", 2nd Edition, Addison-Wesley 2014.
4. E Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw-Hill Education, 2008.
5. Paul Deitel, Harvey Deitel, "C++ How to Program", 8th Edition, Prentice Hall, 2011.
6. John R. Hubbard, "Programming with C++", Schaum's Series, 2nd Edition, 2000.
7. Andrew Koeni, Barbara, E. Moo, "Accelerated C++", Published by Addison-Wesley, 2000.
8. Scott Meyers, "Effective C++", 3rd Edition, Published by Addison-Wesley, 2005.
9. Harry, H. Chaudhary, "Head First C++ Programming: The Definitive Beginner's Guide", First Create space Inc, O-D Publishing, LLC USA.2014
10. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
11. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by AddisonWesley, 5th Edition, 2012

PRACTICAL
BCA120C1: LAB: PROGRAMMING FUNDAMENTALS USING C/C++

PRACTICAL: 60 LECTURES (CREDITS: 2)
1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series S = 1+1/2+1/3+1/4+......
4. WAP to compute the sum of the first n terms of the following series
   S =l-2+3-4+5 ....
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers, WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):

$$*$$
$$***$$
$$*****$$
$$*******$$
$$*********$$

10. 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 array
    iv) Print the maximum and minimum element of array
    v) Remove the duplicates from the array
    vi) Print the array in reverse order

    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.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
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 find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() *I* calloc() functions or new operator.

16. Write a menu driven 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) Calculate length of the string (use pointers)
f) Convert all lowercase characters to uppercase
g) Convert all uppercase characters to lowercase
h) Calculate number of vowels
i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i)using recursion, (ii) using iteration
19. WAP to calculate Factorial of a number (i)using recursion, (ii) using iteration
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
a) Sum b) Difference c) Product d) Transpose
22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime Polymorphism).
23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
24. Create a class Box containing length, breath and height. Include following methods in it:
    a) Calculate surface Area
    b) Calculate Volume
    c) Increment, Overload ++ operator (both prefix & postfix)
    d) Decrement, Overload - operator (both prefix & postfix)
    e) Overload operator = (to check equality of two boxes), as a friend function
    f) Overload Assignment operator
    g) Check if it is a Cube or cuboid
Write a program which takes input from the user for length, breath and height to test the above class.
25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
26. Write a program to retrieve the student information from file created in previous question and print it in following format:
Roll No. Name Marks
27. Copy the contents of one text file to another file, after removing all whitespaces.
28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.
****