C++ Assignment

1. Create two functions, swapByValue and swapByAddress, to swap the values of two integers using call by value and call by address, respectively. Show the output before and after swapping for both functions.

2. Define a Rectangle structure with length and width as members. Write two functions, areaByValue and areaByAddress, that calculate the area of a rectangle when passed by value and by address, respectively. Implement a main function to demonstrate the usage of these functions.

3. Create a Person structure with name and age as members. Write a C++ program to dynamically allocate memory for a Person instance, read the name and age from the user, and display the values. Ensure to properly free the allocated memory.

4. Write a C++ program that dynamically allocates memory for an array of integers using malloc, reads n elements from the user, and then displays the elements. Finally, free the allocated memory using free.

5. Create a class Student with name and grade as private members. Implement getter and setter methods for these members. Split the class definition and implementation into separate header (Student.h) and source (Student.cpp) files. Write a main.cpp file to test the class by setting and getting the values of name and grade.

6. Write a program to demonstrate different storage classes in C/C++. Show examples for auto, static, extern, and register storage classes.

7. Write a program to demonstrate the different scopes in C/C++: global scope, local scope, and block scope. Create variables with the same name in different scopes and show how they are accessed.

8. Write a C++ program to use different stream objects (cin, cout, cerr, clog). Show examples of using each stream object for input, standard output, error output, and log messages.

9. Write a program to demonstrate output formatting in C++. Use manipulators like setw, setprecision, and fixed to format the output of a floating-point number.

10. Create a class Animal with name and age as public members. Implement a member function to display the details of the animal. Create an object of the Animal class in the main function, set its attributes, and call the display function.

11. Create an empty class in C++. Write a program to create an object of this empty class and demonstrate that it can be instantiated.

12. Write a C++ program to use this pointer. Create a class Example with a private member variable value and a public member function setValue that uses the this pointer to set the value of the member variable. Also, implement a member function display to print the value.

13. Create a Person class with private members name and age. Implement getter and setter methods. Write a main function to create an object of the Person class, set its attributes using the setter methods, and get the values using the getter methods.

14. Write a C++ program for function overloading. Create two overloaded functions named add, one that adds two integers and another that adds two floating-point numbers.Show how function overloading works by calling both functions in the main function.

15. Write a C++ program to use the default arguments in functions. Create a function display that takes an integer parameter with a default value. Call the function with and without passing an argument and show the output.

16. Create a class Dog with a public member variable name and a member function bark. Write a program to create an object of the Dog class, set its name, and call the bark function.

17. Create an enumeration Color with values RED, GREEN, and BLUE. Write a function displayColor that takes a Color parameter and prints the corresponding color name. Call this function from main function & show how enum works.

18. Write a C++ program which uses the constructors: default constructor, parameterized constructor, and copy constructor. Create a Box class with length

and breadth as members. Implement all three constructors and a member function to display the dimensions of the box. Create objects using each constructor type in the main function.

19. Define a structure Point with x and y as members. Write a program to demonstrate aggregate initialization by initializing a Point object with values for x and y.

20. Create a class Sample with private members a and b. Implement a constructor that uses a member initializer list to initialize these members. Write a member function to display the values of a and b, and test the implementation in the main function.