

C++ Assignments

1. Define a point class, where each object represents point in Cartesian coordinates (x, y). Define objects of this class and calculate the distance between the 2 points through a friend function.
2. Define polar class where each object represents point in polar coordinates (angle and radius). Include static data member which keeps track the number of active objects of the class.
3. Define a complex class, define default constructor, parameterized constructors, copy constructor, destructor. Overload +, -, unary -, ++ (prefix, postfix), =, comma (,), -, << and >> operators.
4. Define string class with dynamic memory allocation for string. Define default constructor, parameterized constructors, copy constructor, destructor, Overload +, [], =, <<, >> operators. Observe the behavior of shallow copying and deep copying.
5. Define matrix class with dynamic memory allocation for 2-D matrix. Include constructors, destructor, overload + (m1+m2, m1+3 & 4+m2), -, << and >> operators. Also overload new and delete operators by using malloc & free functions internally.
6. Define DSSD_STUDENT class with data members rollno, name, marks of different modules, total marks and grade. Overload >> for reading the DSSD_STUDENT details, Overload << for displaying the DSSD_STUDENT details, include a friend function generate_results (DSSD_STUDENT [], int n); which takes an array of DSSD_STUDENT objects corresponding to n students and generates total marks and grade for each of the student. Please display the results of all students using overloaded << operator. Also include functionality to transfer the details of all objects into a file and retrieve the details & print on the screen.
7. Define a Double ended queue class. Define a Stack class by extending the functionality of Double ended queue class. Create objects of both the classes and test the functionality.

8. Define a shape class. Derive Triangle, Square and Rectangle classes. Demonstrate the use of virtual function by including display_area () as a virtual function
9. Demonstrate the use of virtual destructor with an example class hierarchy
10. Demonstrate the use of virtual base class
11. Develop a class hierarchy and demonstrate the usage of constructors in base and derived classes.
12. Develop a class hierarchy and understand the visibility modes of members derived from base class through various derivation (private, public and protected) modes
13. Write a program to display the marks sheet of 10th class students. Use formatted output operations, while displaying the details
14. Define a template stack class. Using this template class, create stack objects of int, float, char* and complex type. Include exception-handling functionality.
15. Define a template double ended queue (deque). Derive template stack and queue classes from it. Test it for int and char* type of objects.