

CPP Problem Design

Subject: Advanced Class Point in plane

Contributor: 鄭博安, 王聖文, 林岳儒

Main testing concept: CONSTRUCTORS AND OTHER TOOLS

Basics

- ☐ C++ BASICS
- ☐ FLOW OF CONTROL
- ☒ FUNCTION BASICS
- ☐ PARAMETERS AND OVERLOADING
- ☐ ARRAYS
- ☒ STRUCTURES AND CLASSES
- ☒ CONSTRUCTORS AND OTHER TOOLS
- ☐ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
- ☐ STRINGS
- ☐ POINTERS AND DYNAMIC ARRAYS

Functions

- ☐ SEPARATE COMPILATION AND NAMESPACES
- ☐ STREAMS AND FILE I/O
- ☐ RECURSION
- ☐ INHERITANCE
- ☐ POLYMORPHISM AND VIRTUAL FUNCTIONS
- ☐ TEMPLATES
- ☐ LINKED DATA STRUCTURES
- ☐ EXCEPTION HANDLING
- ☐ STANDARD TEMPLATE LIBRARY
- ☐ PATTERNS AND UML

Description:

The type Point is a fairly simple data type, but under another name (the template class pair) this data type is defined and used in the C++ Standard Template Library, although you need not know anything about the Standard Template Library to do this exercise. Write a definition of a class named Point that might be used to store and manipulate the location of a point in the plane. You will need to declare and implement the following member functions:

- a. A member function set that sets the private data after an object of this class is created. Attention! The data must be an array of float in size 2, initialized by zero.
- b. A member function to move the point by an amount along the horizontal and vertical directions specified by the first and second arguments.
- c. A member function to rotate the point by 90 degrees clockwise around the origin.
- d. A member function to reflect point in the origin.
- e. Two const inspector functions to retrieve the current coordinates of the point (horizontal and vertical).
- f. A copy constructor used to copy value of other point.
- g. A destructor used to release memory before the instance of class destroyed. Avoiding memory leak!

Document these functions with appropriate comments. Embed your class in a test program that requests data for several points from the user, creates the points, then exercises the member functions.

Input:

main.in

Output:

Sample Input / Output :

Sample Input	Sample Output
<pre>int main(void) { Point point; point.Set(0, 5); cout << point.RetrieveHorizontal() << " " << point.RetrieveVertical() << endl; point.Reflect(); cout << point.RetrieveHorizontal() << " " << point.RetrieveVertical() << endl; point.Move(1, 2); cout << point.RetrieveHorizontal() << " " << point.RetrieveVertical() << endl; point.Rotate(); cout << point.RetrieveHorizontal() << " " << point.RetrieveVertical() << endl; Point copy = point; cout << "Copy point:" << endl; cout << copy.RetrieveHorizontal() << " " << copy.RetrieveVertical() << endl; return 0; }</pre>	<pre>0 5 0 -5 1 -3 -3 -1 Copy point: -3 -1</pre>

- ☐ Easy, Only basic programming syntax and structure are required.
- ☒ Medium, Multiple programming grammars and structures are required.
- ☐ Hard, Need to use multiple program structures or more complex data types.

Expected solving time:

20minutes

Other notes: