**CPP Problem Design**

|  |
| --- |
| **Subject:** **Advanced Class Point** |
| **Contributor: 鄭博安, 王聖文, 林岳儒** |
| **Main testing concept:** CONSTRUCTORS AND OTHER TOOLS   |  |  | | --- | --- | | **Basics** | **Functions** | | □ 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 | □ 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:   1. 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. 2. A member function named **Move** to move the point along the horizontal and vertical directions specified by the first and second arguments. 3. A member function named **Rotate** to rotate the point by 90 degrees clockwise around the origin. 4. A member function named **Reflect** to reflect point in the origin. 5. Two **const getter** functions to retrieve the current coordinates of the point (horizontal and vertical). 6. A **copy constructor** used to copy value of other point. 7. A **destructor** used to release memory before the instance of class destroyed. Avoiding memory leak!   **Input:**  沒有輸入，實作Point.h, Point.cpp  **Output:**  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | #include<iostream>  #include"Point.h"  using namespace std;  int main(void) {  Point point;  point.Set(0, 5);  cout << point. GetHorizontal() << " " << point. GetVertical() << endl;  point.Reflect();  cout << point. GetHorizontal() << " " << point. GetVertical() << endl;  point.Move(1, 2);  cout << point. GetHorizontal() << " " << point. GetVertical() << endl;  point.Rotate();  cout << point.GetHorizontal() << " " << point.GetVertical() << endl;  Point copy = point;  cout << "Copy point:" << endl;  cout << copy. GetHorizontal() << " " << copy. GetVertical() << endl;  return 0;  } | 0 5  0 -5  1 -3  -3 -1  Copy point:  -3 -1 | |
| **□ Eazy,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:**  **PI** |