**CPP Problem Design**

|  |
| --- |
| **Subject:** **Class Point in plane** |
| **Contributor: 陳俊儒, 林承達, 廖宣瑋** |
| **Main testing concept:** Class member function   |  |  | | --- | --- | | **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:**  Please implement a class named **Point** to store and manipulate the position of the point on the screen.   * Class **Point** has two private integer variables: **vertical** and **horizontal**, where **vertical** is the x coordinate and **horizontal** is the y coordinate of a point. * The default coordinate of **Point** when constructing is (0,0). * Please implement the following member functions: * **void Set(int vertical, int horizontal)**   Reset the coordinate position of the point by the given the x and y.   * **void Move(int x, int y)**   Move the current point **x** pixels on the x-axis and **y** pixels on the y-axis.   * **void Rotate()**   Rotate this point 90 degrees clockwise from the origin.   * **int RetrieveVertical() const**   Get the value y of the point.   * **int RetrieveHorizontal() const**   Get the value x of the point.  **Input:**  The **main()** function in your submission will be replaced when judging.  You can use the **main()** function in the Sample Input to test your program.  This exercise does not have an input.  **Output:**  The result of executing your program with the given main function.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | int main(void) {  Point point;  point.Set(0, 5);  cout << point.RetrieveVertical() << " " << point.RetrieveHorizontal() << endl;  point.Move(1, 2);  cout << point.RetrieveVertical() << " " << point.RetrieveHorizontal() << endl;  for (int i = 0; i < 4; i++) {  point.Rotate();  cout << point.RetrieveVertical() << " " << point.RetrieveHorizontal() << endl;  }  return 0;  } | 0 5  1 7  7 -1  -1 -7  -7 1  1 7 | |
| **□ 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:**  20 minutes |
| **Other notes:** |