**CPP Problem Design Example**

|  |
| --- |
| **Subject: Shapes** |
| **Contributor: 鍾賢廣** |
| **Main testing concept: structure/class**   |  |  | | --- | --- | | **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:**  Define **Shape** object which inherits by **Triangle**, **Circle**, and **Rectangle** object.  The Shape object has the following functions:  virtual float GetArea(); return total area the shape has.  virtual float GetPerimeter(); return boundary length of the shape.  virtual void GetCenter(float\* x, float\* y); return the position of the center of mass of the shape  static bool IsOverlap(shape\* a, shape\* b); check if two shapes overlap.  The Circle should have the following constructor and functions:  Circle(float r, float x, float y); create a circle by given a radius and 2-d position.  The Triangle should have the following constructor and functions:  Triangle(float x1, float y1, float x2, float y2, float x3, float y3); create a triangle by given three 2-d positions represented as the three points of the triangle.  The Rectangle should have the following constructor and functions:  Rectangle(float left, float right, float top ,float down); create a rectangle by given the axis of the four sides(left-right: x-axis, top-down: y-axis).  **Input:**  There are no further inputs, please refers to the example.cpp files.  **The name of the object and files please refer to the example.cpp file.**  **Output:**  Please refer to the sample output  **Sample Input / Output：**   |  |  | | --- | --- | | example.cpp | Sample Output | | Please refer to “example.cpp” | The Area of Shape 0: 50.2655  The Perimeter of Shape 0: 25.1327  The Center of Shape 0: ( -3, -2 )  The Area of Shape 1: 36.3168  The Perimeter of Shape 1: 21.3628  The Center of Shape 1: ( 1.5, 3 )  The Area of Shape 2: 26.25  The Perimeter of Shape 2: 24.2186  The Center of Shape 2: ( 5.5, 0 )  The Area of Shape 3: 3.075  The Perimeter of Shape 3: 10.9814  The Center of Shape 3: ( -1.83333, 0.166667 )  The Area of Shape 4: 79.75  The Perimeter of Shape 4: 36.5  The Center of Shape 4: ( 0.5, -4.375 )  The Area of Shape 5: 24  The Perimeter of Shape 5: 20  The Center of Shape 5: ( 2, -4 )  Shape 1 and 0 is overlap.  Shape 2 and 1 is overlap.  Shape 3 and 0 is overlap.  Shape 3 and 1 is overlap.  Shape 4 and 0 is overlap.  Shape 4 and 2 is overlap.  Shape 5 and 0 is overlap.  Shape 5 and 2 is overlap.  Shape 5 and 4 is overlap. | |
| **□ 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 complex data types.** |
| **Expected solving time:**  25 minutes |
| **Other notes:**  You can define any amount of functions and parameters in your object.  The floating point values will be output according to default output settings namely you did not need to change the output settings.  Two shapes kissing each other does not count as an overlap.  Here are shapes in the example input. |