**BJ1000. Circle and Rectangle**

**Description**

以点（Point）类为基类，重新派生类 Rectangle、Circle。坐标原点为(0,0)，矩形水平放置，由左下方的顶点和长宽定义。

圆由圆心和半径定义。派生类操作判断任一坐标点是在图形内，还是在图形的边缘上，还是在图形外。

缺省初始化图形退化为坐标原点（0，0）

要求拷贝构造函数和定义默认构造函数

编程测试类设计是否正确。

class Point{

private:

double x,y;

public:

Point();

Point(double xv,double yv);

Point(Point& pt);

double getx();

double gety();

};

类Circle 和类Rectangle 都包含成员函数：int position(Point &pt)。

假如点A落在圆内或是矩形内，则该函数返回-1；如果落在圆上或是矩形上，则返回0,；如果落在外面，则返回1。

主函数：

int main()

{

Circle cc1(3,4,5);

Rectangle rt1(0,0,6,8);

Point p1(0,0);

cout<<"point p1:";

switch(rt1.position(p1))

{

 case 0:cout<<"on-rectangle"<<endl;break;

 case -1:cout<<"inside-rectangle"<<endl;break;

 case 1:cout<<"outside-rectangle"<<endl;break;

}

switch(cc1.position(p1))

{

 case 0:cout<<"on-circle"<<endl;break;

 case -1:cout<<"inside-circle"<<endl;break;

 case 1:cout<<"outside-circle"<<endl;break;

}

return 0;

}

**Input**

 none

**Output**

 none

**Hint**

  const double PI=3.14;

 you only need to submit the class:Point 、Circle、Rectangle

**BJ1001. The Person, Student, Employee, Faculty and Staff Class**

**Description**

Design a class named **Person** and its two derived classes named **Student** and **Employee**. Make **Faculty** and **Staff** derived classes of **Employee**.  
Person class has the following member:  
  string name;  
  string address;  
  string phoneNumber;  
  string email;  
Student class has the following member:  
  enum class\_status{frssman,sophomore,junior,senior};    
  class\_status status;  
Employee class has the following member:  
  string office;  
  int salary;  
  MyDate dateHired;  
Faculty class has the following member:  
  string officeHours;  
  int rank;  
Staff class has the following member:  
  string title;  
MyDate class contains the following member:   
  int year;  
  int month;  
  int day;  
  
Override the toString function in each class to display the class name.  
For example, in Employee class:  
string toString()  
{  
  return "Employee";  
}

Make sure that the following code:   
  void f(Person &p)  
  {  
    cout << p.toString() << endl;  
  }

  Person person;  
  Student student;  
  Employee employee;  
  Faculty faculty;  
  Staff staff;  
  f(person);  
  f(student);  
  f(employee);  
  f(faculty);  
  f(staff);  
  
output:  
  Person  
  Student  
  Employee  
  Faculty  
  Staff

**BJ1002. The Triangle Class**

**Description**

Design a class named Triangle that extends **GeometricObject** class.  
The class contains:  
Three **double** data fields named **side1**, **side2**, and **side3** with default values **1.0** to denote three sides of the triangle.  
A no-arg constructor that creates a default triangle with color = "blue", filled = true.  
A constructor that creates a triangle with the specified side1, side2, side3 and color = "blue", filled = true.  
The accessor functions for all three data fields, named **getSide1()**, **getSide2()**, **getSide3()**.  
A function named **getArea()** that returns the area of this triangle.  
A function named **getPerimeter()** that returns the perimeter of the triangle.  
  
class GeometricObject  
{  
public:  
  GeometricObject(string color, bool filled)  
  {  
    this->color = color;  
    this->filled = filled;  
  }  
    
  string getColor()  
  { return color; }  
  void setColor(string color)  
  { this->color = color; }  
  bool isFilled()  
  { return filled; }  
  void setFilled(bool filled)  
  { this->filled = filled;}  
  string toString()  
  {  
    return "Geometric object color " + color +  
    " filled " + ((filled) ? "true" : "false");  
  }  
private:  
  string color;  
  bool filled;  
};

**Hint**

Please submit the GeometricObject class and Triangle class.

Don't submit the main() function.

**BJ1003. Rectangle**

**Description**

1、定义基类 Point，使其能够初始化坐标（x，y），移动坐标，返回坐标

2、定义派生类 Rectangle，有成员变量 width、high；并且有成员函数，使其能够返回矩形相关属性：宽、高、面积

基类：

class Point

{

    int x,y;

public:

    Point(int x,int y);

    void Move(int x,int y);

    int Getx();

    int Gety();

};

主函数：

cin>>x>>y;  
        cin>>length>>width;  
        Rectangle R(x,y,length,width);  
        cin>>xmove>>ymove;  
        R.Move(xmove,ymove);  
        cout <<R.Getx()<<" "<<R.Gety()<<" "  
             <<R.Getlength()<<" "<<R.Getwidth()  
              <<" "<<R.Getarea()<<endl;

**Input**

第一行输入n，代表n组测试例子

第二行输入x，y ，表示坐标原点A（x,y）

第三行输入length，width，表示矩形长、宽

第四行输入xmove，ymove，表示原点A在横纵坐标移动的距离。

**Output**

每组数据，输出一行，分别是：移动后的横坐标、纵坐标；矩形长、宽；面积。

**Sample Input**

http://soj.sysu.edu.cn/images/clipboard.jpgCopy sample input to clipboard

1

0 0

8 4

10 12

**Sample Output**

10 12 8 4 32

**Hint**

you ﻿only need  to submit the class Point and Rectangle.

方法名首字母大写,其余字母小写

**BJ1004. Constructor and inheritance**

**Description**

Class A is defined as follows:

class A  
{  
public:  
    A(int a0):a(a0) {}  
    void show() { cout<<"a="<<a<<endl; }  
private:  
    int a;  
};

Class B inherits from A. Its constructor has two parameters, with the first one for the member a and the second for the member b. Class B also redefines the show function. Now you need to complete the definition of class B to produce the desired output.

class B  
{  
public:  
    B(int, int);  
    void show();  
private:  
    int b;  
};

Your submitted source code should include the whole implementation of the class B, but without the class A.

No main() function should be included.

**Note**: the main() function of the test framework looks like this:

------------------------------------------------------------------------------

int main()  
{  
    B b(1, 2);  
    b.show();  
    return 0;  
}

------------------------------------------------------------------------------

**Sample Output**

a=1

b=2