Course: ENSF 614 — Fall 2023
Lab 5
Instructor: Mahmood Moussavi
Student Name: Brandon Lac
Submission Date: October 21, 2023

Exercice A

Code for Point.h

```
// Point.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Date of submission: October 21 2023

#ifndef Point_H
#define Point_H

class Point
{
private:
    double x;
    double y;
    int id;
    static int count; // Static variable to keep track of how many points have
been created

public:
    Point(double x, double y);
    // Constructor
    static int counter();
    // PROMISES: to return the number of objects of class Point at anytime.

    static double distance(Point &p, Point &op);
    // PROMISES: to return the distance between 2 points.

    double distance(Point &p);
    // PROMISES: to return the distance between 2 points.

    void display() const;
    // PROMISES: to display the x and y coordinates in the following format:
    // X-coordinate: ######.##
    // Y-coordinate: ######.##

    double getx() const;
    // PROMISES to retrieve the x value of point

    int getid() const;
    // PROMISES to retrieve the id value of point

    double gety() const;
```

```
    // PROMISES to retrieve the y value of point

    void setx(double x);
    // PROMISES to set the x value of point

    void sety(double y);
    // PROMISES to set the y value of point
};
#endif
```

Code for Point.cpp

```
//  Point.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Date of submission: October 21 2023
#include "Point.h"
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

int Point::count = 0;
Point::Point(double x, double y)
{

    this->x = x;
    this->y = y;
    count++;
    id = 1000 + count;
}

void Point::display() const
{
    cout
        << "X-coordinate: " << setw(9) << setfill('0') << std::fixed <<
std::setprecision(2) << x << endl;
    cout << "Y-coordinate: " << setw(9) << setfill('0') << std::fixed <<
std::setprecision(2) << y << endl;
}

int Point::counter()
{
```

```
    return count;
}

int Point::getid() const
{
    return id;
}

double Point::getx() const
{
    return x;
}

double Point::gety() const
{
    return y;
}

void Point::setx(double x)
{

    this->x = x;
}

void Point::sety(double y)
{

    this->y = y;
}

double Point::distance(Point &p)
{
    return (pow(pow(abs(p.x - x), 2) + pow(abs(p.y - y), 2), 0.5));
}

double Point::distance(Point &p, Point &op)
{
    return (pow(pow(abs(p.getx() - op.getx()), 2) + pow(abs(p.gety() -
op.gety()), 2), 0.5));
}
```

Output Testing for Point

```
This program was created by Brandon Lac
The count for the number of point objects is 1
ID of the first point m is : 1001
ID of the second point n is : 1002
The count for the number of point objects after creating 2 objects: 2

Expected to dispaly the distance between m and n is: 3
The distance between m and n is: 3
Expected second version of the distance function also print: 3
The distance between m and n is again: 3
Displaying point m
X-coordinate: 000006.00
Y-coordinate: 000008.00

Displaying point n
X-coordinate: 000009.00
Y-coordinate: 000008.00
```

Code for Square.cpp

```
// Square.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Submission Date: October 21 2023
#include "Square.h"
#include <cstring>
#include <iostream>
#include "Shape.h"
using namespace std;

Square::Square(double x, double y, double side, const char *s) : Shape(x, y, s)
{
    side_a = side;
}

double Square::area() const
{
    return side_a * side_a;
}
// PROMISES: returns the area of the sqaure.

double Square::perimeter() const
{
    return 4 * side_a;
}
// PROMISES: returns the perimeter of the square.

void Square::set_side_a(double side)
{
    this->side_a = side;
}
// PROMISES: sets the side_a

double Square::getSideA() const
{
    return side_a;
}
// PROMISES: returns the value of side_a.

void Square::display() const
{
    cout << "Square Name: " << getName() << endl;
    cout << "X-coordinate: " << origin.getx() << endl;
```

```
    cout << "Y-coordinate: " << origin.gety() << endl;
    cout << "side a: " << side_a << endl;
    cout << "Area: " << this->area() << endl;
    cout << "perimeter: " << this->perimeter() << endl;
}
// PROMISES:  that prints on the screen the Square's name, x and y coordinates
//            of point origin,side a, area, and perimeter in the following
format:
//            Shape Name:
//            X-coordinate:
//            Y-coordinate:
//            side a:
//            Area:
//            Perimeter
```

Code for Square.h

```
// Shape.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Submission Date: October 21 2023
#ifndef Shape_H
#define Shape_H
#include "Point.h"
class Shape
{
public:
    Shape(double x, double y, char const *n);
    // Constructor

    virtual ~Shape();
    // Deconstructor

    Point getOrigin() const;
    // PROMISES: the object origin which is of type Point.

    char *getName() const;
    // PROMISES: returns the name of the shape.

    virtual void display() const;
    // PROMISES:  that prints on the screen the shape's name, x and y coordinates
    //            of point origin, in the following format:
```

```
    //            Shape Name:
    //            X-coordinate:
    //            Y-coordinate:

    virtual double area() const = 0;

    virtual double perimeter() const = 0;

    double distance(Shape &other) const;
    // PROMISES: Returns the distance between this shape and other.

    static double distance(Shape &the_shape, Shape &other);
    // PROMISES: Returns the distance between two shapes provided by
    // the_shape and other.

    void move(double dx, double dy);
    // PROMISES: Changes the position of the shape, the current x and y
coordinates to
    // x+dx, and y+dy.

protected:
    Point origin;
    char *shapeName;
};

#endif
```

Shape.cpp

```
// Shape.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Submission Date: October 21 2023
#include "Shape.h"
#include <cstring>
#include <string>
#include <iostream>
#include <cmath>
#include "Point.h"
using namespace std;
Shape::Shape(double x, double y, char const *n) : origin(x, y)
{
    this->shapeName = new char[sizeof(n)];
    strcpy(shapeName, n);
```

```
}

Shape::~Shape()
{
    delete[] shapeName;
}

Point Shape::getOrigin() const
{
    return origin;
}

char *Shape::getName() const
{
    return this->shapeName;
}

void Shape::display() const
{
    cout << "Shape Name: " << this->shapeName << endl;
    cout << "X-coordinate: " << this->origin.getx() << endl;
    cout << "Y-coordinate: " << this->origin.gety() << endl;
}

double Shape::distance(Shape &other) const
{
    return (pow(pow(abs(other.origin.getx() - origin.getx()), 2) +
pow(abs(other.origin.gety() - origin.gety()), 2), 0.5));
}

double Shape::distance(Shape &the_shape, Shape &other)
{
    return (pow(pow(abs(other.origin.getx() - the_shape.origin.getx()), 2) +
pow(abs(other.origin.gety() - the_shape.origin.gety()), 2), 0.5));
}

void Shape::move(double dx, double dy)
{
    origin.setx(origin.getx() + dx);
    origin.sety(origin.gety() + dy);
}
```

Shape.h

```
// Shape.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Submission Date: October 21 2023
#ifndef Shape_H
#define Shape_H
#include "Point.h"
class Shape
{
public:
    Shape(double x, double y, char const *n);
    // Constructor

    virtual ~Shape();
    // Deconstructor

    Point getOrigin() const;
    // PROMISES: the object origin which is of type Point.

    char *getName() const;
    // PROMISES: returns the name of the shape.

    virtual void display() const;
    // PROMISES:  that prints on the screen the shape's name, x and y coordinates
    //            of point origin, in the following format:
    //            Shape Name:
    //            X-coordinate:
    //            Y-coordinate:

    virtual double area() const = 0;

    virtual double perimeter() const = 0;

    double distance(Shape &other) const;
    // PROMISES: Returns the distance between this shape and other.

    static double distance(Shape &the_shape, Shape &other);
    // PROMISES: Returns the distance between two shapes provided by
    // the_shape and other.

    void move(double dx, double dy);
```

```
    // PROMISES: Changes the position of the shape, the current x and y
coordinates to
    // x+dx, and y+dy.

protected:
    Point origin;
    char *shapeName;
};

#endif
```

Output Testing for Square and the move function of Shape

```
This program was created by Brandon Lac


Testing Functions in class Square:
Square Name: SQUARE - S
X-coordinate: 5
Y-coordinate: 7
side a: 12
Area: 144
perimeter: 48

 after moving the x and y by +2
Square Name: SQUARE - S
X-coordinate: 7
Y-coordinate: 9
side a: 12
Area: 144
perimeter: 48
```

Rectangle.cpp

```
// Rectangle.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Submission Date: October 21 2023
#include "Rectangle.h"
#include <string>
#include <iostream>
#include "Square.h"
using namespace std;

Rectangle::Rectangle(double x, double y, double side_a,
                     double side_b, char const *n) : Shape(x, y, n), Square(x, y,
side_a, n), side_b(side_b)
{
}
double Rectangle::area() const
{
    return (this->side_a * this->side_b);
}
// PROMISES: returns the area of the rectangle.
double Rectangle::perimeter() const
{
    return (side_a * 2 + side_b * 2);
}
// PROMISES: returns the perimeter of the rectangle.
void Rectangle::set_side_b(double b)
{
    side_b = b;
}
// PROMISES: sets the value of side_b.
double Rectangle::getSideB() const
{
    return side_b;
}
// PROMISES: returns the value of side_b.
void Rectangle::display() const
{
    cout << "Rectangle Name: " << shapeName << endl;
    cout << "X-coordinate: " << origin.getx() << endl;
    cout << "Y-coordinate: " << origin.gety() << endl;
    cout << "side a: " << side_a << endl;
```

```
    cout << "side b: " << side_b << endl;
    cout << "Area: " << area() << endl;
    cout << "perimeter: " << perimeter() << endl;
}
// PROMISES:  that prints on the screen the Rectangle’s name, x and y coordinates
//            of point origin,side a,side_b, area, and perimeter in the following
format:
//            Square Name:
//            X-coordinate:
//            Y-coordinate:
//            side a:
//            Area:
//            Perimeter
```

Rectangle.h

```
// Rectangle.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A
// Created by: Brandon Lac
// Submission Date: October 21 2023
#ifndef Rectangle_H
#define Rectangle_H
#include "Square.h"
class Rectangle : public Square
{
public:
    Rectangle(double x, double y, double side_a,
              double side_b, char const *n);
    // Constructor
    double area() const;
    // PROMISES: returns the area of the rectangle.
    double perimeter() const;
    // PROMISES: returns the perimeter of the rectangle.
    void set_side_b(double b);
    // PROMISES: sets the value of side_b.
    double getSideB() const;
    // PROMISES: returns the value of side_b.
    void display() const;
    // PROMISES:  that prints on the screen the Rectangle’s name, x and y
coordinates
    //            of point origin,side a,side_b, area, and perimeter in the
following format:
    //            Rectangle Name:
```

```
    //            X-coordinate:
    //            Y-coordinate:
    //            side a:
    //            Area:
    //            Perimeter

protected:
    double side_b;
};

#endif
```

Testing of Rectangle

```
This program was created by Brandon Lac

Testing Functions in class Rectangle:Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 12
side b: 15
Area: 180
perimeter: 54
Rectangle Name: RECTANGLE B
X-coordinate: 16
Y-coordinate: 7
side a: 8
side b: 9
Area: 72
perimeter: 34

Distance between square a, and b is: 11
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 12
side b: 15
Area: 180
perimeter: 54

Testing assignment operator in class Rectangle:
Rectangle Name: RECTANGLE rec2
X-coordinate: 3
Y-coordinate: 4
side a: 11
side b: 7
Area: 77
perimeter: 36

Expected to display the following values for objec rec2:
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
Side a: 12
Side b: 15
Area: 180
Perimeter: 54

If it doesn't there is a problem with your assignment operator.

Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 12
side b: 15
Area: 180
perimeter: 54

Testing copy constructor in class Rectangle:
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 100
side b: 200
Area: 20000
perimeter: 600

Expected to display the following values for objec rec2:
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
Side a: 100
Side b: 200
Area: 20000
Perimeter: 600

If it doesn't there is a problem with your assignment operator.

Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 100
side b: 200
Area: 20000
perimeter: 600
```

Testing of Arrays of pointers and polymorphism

```
Testing array of pointers and polymorphism:
Square Name: SQUARE - S
X-coordinate: 7
Y-coordinate: 9
side a: 12
Area: 144
perimeter: 48
Rectangle Name: RECTANGLE B
X-coordinate: 16
Y-coordinate: 7
side a: 8
side b: 9
Area: 72
perimeter: 34
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 12
side b: 15
Area: 180
perimeter: 54
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 100
side b: 200
Area: 20000
perimeter: 600
```

GraphicsWord.cpp

```cpp
// GraphicsWorld.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A & B
// Created by: Brandon Lac
// Submission Date: October 21 2023
#include "GraphicsWorld.h"
#include "Point.h"
#include <iostream>
#include "Square.h"
#include "Rectangle.h"
#include "Circle.h"
#include "CurveCut.h"
using namespace std;
// GraphicsWorld.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A & B
// Created by: Brandon Lac
// Submission Date: October 21 2023
void GraphicsWorld::run()
{
     cout << "This program was created by Brandon Lac" << endl;
```

```
#if 0  // Change 0 to 1 to test Point
     Point m(6, 8);
     cout << "The count for the number of point objects is " << m.counter() <<
endl;
     cout << "ID of the first point m is : " << m.getid() << endl;
     Point n(6, 8);
     cout << "ID of the second point n is : " << n.getid() << endl;
     cout << "The count for the number of point objects after creating 2 objects:
" << m.counter() << endl;
     n.setx(9);
     cout << "\nExpected to dispaly the distance between m and n is: 3";
     cout << "\nThe distance between m and n is: " << m.distance(n);
     cout << "\nExpected second version of the distance function also print: 3";
     cout << "\nThe distance between m and n is again: "
          << Point::distance(m, n) << endl;
     cout << "Displaying point m" << endl;
     m.display();
     cout << endl;
     cout << "Displaying point n" << endl;
     n.display();
#endif // end of block to test Point
#if 0  // Change 0 to 1 to test Square
     cout << "\n\nTesting Functions in class Square:" << endl;
     Square s(5, 7, 12, "SQUARE - S");
     s.display();
     s.move(2, 2);
     cout << "\nAfter moving the x and y by +2\n";
     s.display();
#endif // end of block to test Square
#if 1  // Change 0 to 1 to test Rectangle
     cout << "\nTesting Functions in class Rectangle:";
     Rectangle a(5, 7, 12, 15, "RECTANGLE A");
     a.display();
     Rectangle b(16, 7, 8, 9, "RECTANGLE B");
     b.display();
     double d = a.distance(b);
     cout << "\nDistance between square a, and b is: " << d << endl;
     Rectangle rec1 = a;
     rec1.display();
     cout << "\nTesting assignment operator in class Rectangle:" << endl;
     Rectangle rec2(3, 4, 11, 7, "RECTANGLE rec2");
     rec2.display();
     rec2 = a;
     a.set_side_b(200);
     a.set_side_a(100);
```

```
     cout << "\nExpected to display the following values for objec rec2: " <<
endl;
     cout << "Rectangle Name: RECTANGLE A\n"
          << "X-coordinate: 5\n"
          << "Y-coordinate: 7\n"
          << "Side a: 12\n"
          << "Side b: 15\n"
          << "Area: 180\n"
          << "Perimeter: 54\n";
     cout << "\nIf it doesn't there is a problem with your assignment
operator.\n"
          << endl;
     rec2.display();
     cout << "\nTesting copy constructor in class Rectangle:" << endl;
     Rectangle rec3(a);
     rec3.display();
     a.set_side_b(300);
     a.set_side_a(400);
     cout << "\nExpected to display the following values for objec rec2: " <<
endl;
     cout << "Rectangle Name: RECTANGLE A\n"
          << "X-coordinate: 5\n"
          << "Y-coordinate: 7\n"
          << "Side a: 100\n"
          << "Side b: 200\n"
          << "Area: 20000\n"
          << "Perimeter: 600\n";
     cout << "\nIf it doesn't there is a problem with your assignment
operator.\n"
          << endl;
     rec3.display();
#endif // end of block to test Rectangle
#if 0  // Change 0 to 1 to test using array of pointer and polymorphism
     cout << "\nTesting array of pointers and polymorphism:" << endl;
     Shape *sh[4];
     sh[0] = &s;
     sh[1] = &b;
     sh[2] = &rec1;
     sh[3] = &rec3;
     sh[0]->display();
     sh[1]->display();
     sh[2]->display();
     sh[3]->display();
#endif // end of block to test array of pointer and polymorphism
```

```
#if 0
     cout << "\nTesting Functions in class Circle:" << endl;
     Circle c(3, 5, 9, "CIRCLE C");
     c.display();
     cout << "the area of " << c.getName() << " is: " << c.area() << endl;
     cout << "the perimeter of " << c.getName() << " is: " << c.perimeter() <<
endl;
     d = a.distance(c);
     cout << "\nThe distance between rectangle a and circle c is: " << d;
     CurveCut rc(6, 5, 10, 12, 9, "CurveCut rc");
     rc.display();
     cout << "the area of " << rc.getName() << " is: " << rc.area();
     cout << "the perimeter of " << rc.getName() << " is: " << rc.perimeter();
     d = rc.distance(c);
     cout << "\nThe distance between rc and c is: " << d;
     // Using array of Shape pointers:
     sh[0] = &s;
     sh[1] = &a;
     sh[2] = &c;
     sh[3] = &rc;
     sh[0]->display();
     cout << "\nthe area of " << sh[0]->getName() << "is: " << sh[0]->area() <<
endl;
     cout << "\nthe perimeter of " << sh[0]->getName() << " is: " << sh[0]-
>perimeter() << endl;
     sh[1]->display();
     cout << "\nthe area of " << sh[1]->getName() << "is: " << sh[1]->area() <<
endl;
     cout << "\nthe perimeter of " << sh[0]->getName() << " is: " << sh[1]-
>perimeter() << endl;
     sh[2]->display();
     cout << "\nthe area of " << sh[2]->getName() << "is: " << sh[2]->area();
     cout << "\nthe circumference of " << sh[2]->getName() << " is: " << sh[2]-
>perimeter() << endl;
     sh[3]->display();
     cout << "\nthe area of " << sh[3]->getName() << " is: " << sh[3]->area() <<
endl;
     cout << "\nthe perimeter of " << sh[3]->getName() << " is: " << sh[3]-
>perimeter() << endl;
     cout << "\nTesting copy constructor in class CurveCut:" << endl;
     CurveCut cc = rc;
     cc.display();
     cout << "\nTesting assignment operator in class CurveCut:" << endl;
     CurveCut cc2(2, 5, 100, 12, 9, "CurveCut cc2");
     cc2.display();
```

```
    cc2 = cc;
    cc2.display();
#endif
}
```

GraphicsWorld.h

```
// GraphicsWorld.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE A & B
// Created by: Brandon Lac
// Submission Date: October 21 2023
#ifndef GraphicsWorld_H
#define GraphicsWorld_H
class GraphicsWorld
{
public:
    void run();
    // PROMISES: It has outputs to correctly test if the above classes are
working.
    //           Also prints the name of the author as well.
};

#endif
```

Main file

```
#include <iostream>
#include "GraphicsWorld.h"
int main()
{
    GraphicsWorld b = GraphicsWorld();
    b.run();
}
```

Exercice B

Circle.h

```
// Circle.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE B
// Created by: Brandon Lac
// Submission Date: October 21 2023
#ifndef Circle_H
#define Circle_H
#include "Shape.h"
class Circle : virtual public Shape
{
public:
    Circle(double x, double y, double r, const char *n);
    // Constructor of the circle class
    double area() const override;
    // PROMISES: to return the area of the circle.

    double perimeter() const override;
    // PROMISES: to return the perimeter of the circle.

    double getRadius() const;
    // PROMISES: to return the radius of the circle.

    void setRadius(double r);
    // PROMISES: to set the radius of the circle.

    void display();
    // PROMISES: to display circle's properties in the following format:
    //            Circle Name:
    //            X-coordinate:
    //            Y-coordinate:
    //            Radius:
    //            Area:
    //            Perimeter

protected:
    double radius;
};
#endif
```

Circle.cpp

```cpp
// Circle.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE B
// Created by: Brandon Lac
// Submission Date: October 21 2023
#include "Circle.h"
using namespace std;
#include <iostream>
#include <math.h>
#include "Shape.h"

Circle::Circle(double x, double y, double r, const char *n) : Shape(x, y, n)
{
    radius = r;
}
double Circle::area() const
{
    return (M_PI * pow(radius, 2));
}
double Circle::perimeter() const
{
    return (M_PI * 2 * radius);
}

double Circle::getRadius() const
{
    return radius;
}

void Circle::setRadius(double r)
{
    radius = r;
}

void Circle::display()
{
    cout << "Circle Name: " << shapeName << endl;
    cout << "X-coordinate: " << origin.getx() << endl;
    cout << "Y-coordinate: " << origin.gety() << endl;
    cout << "Radius: " << radius << endl;
    cout << "Area: " << area() << endl;
    cout << "perimeter: " << perimeter() << endl;
}
```

CurveCut.cpp

```
// CurveCut.cpp
// ENSF 614 Fall 2022 LAB 5 - EXERCISE B
// Created by: Brandon Lac
// Submission Date: October 21 2023
#include "Circle.h"
#include "Rectangle.h"
#include "CurveCut.h"
using namespace std;
#include <iostream>
#include <math.h>
CurveCut::CurveCut(double x, double y, double width, double length, double r,
const char *n) : Shape(x, y, n), Circle(x, y, r, n), Rectangle(x, y, width,
length, n)
{
    if (r > length || r > width)
    {
        cout << " Error, radius of circle cannot be greater than the length or
width of the Rectangle";
        exit(0);
    }
}

double CurveCut::area() const
{
    return ((side_a * side_b - 0.25 * (pow(radius, 2) * M_PI)));
}
// PROMISES: to return the area of the rectangle minus the cut

double CurveCut::perimeter() const
{
    return (side_a * 2 + side_b * 2 - 2 * radius + (0.5 * M_PI * radius));
}
// PROMISES: to return the perimeter of the rectangle with the cur

void CurveCut::display()
{
    cout << "CurveCut Name: " << Circle::getName() << endl;
    cout << "X-coordinate: " << Circle::origin.getx() << endl;
    cout << "Y-coordinate: " << Circle::origin.gety() << endl;
    cout << "Radius: " << radius << endl;
    cout << "Area: " << area() << endl;
    cout << "perimeter: " << perimeter() << endl;
}
```

```
// PROMISES: to print the curvecut in the follow format:
// CurveCut Name:
// X-coordinate:
// Y-coordinate:
// Width:
// Length:
// Radius of the cut.
```

CurveCut.h

```
// CurveCut.h
// ENSF 614 Fall 2022 LAB 5 - EXERCISE B
// Created by: Brandon Lac
// Submission Date: October 21 2023
#ifndef CurveCut_H
#define CurveCut_h
#include "Circle.h"
#include "Rectangle.h"

class CurveCut : public Circle, public Rectangle
{
public:
    CurveCut(double x, double y, double width, double length, double r, const
char *n);
    // Constructor of class

    double area() const override;
    // PROMISES: to return the area of the rectangle minus the cut

    double perimeter() const override;
    // PROMISES: to return the perimeter of the rectangle with the cur

    void display();
    // PROMISES: to print the curvecut in the follow format:
    // CurveCut Name:
    // X-coordinate:
    // Y-coordinate:
    // Width:
    // Length:
    // Radius of the cut.
};
#endif
```

Output for Testing of Circle and CurveCut

```
Testing Functions in class Circle:
Circle Name: CIRCLE C
X-coordinate: 3
Y-coordinate: 5
Radius: 9
Area: 254.469
perimeter: 56.5487
the area of CIRCLE C is: 254.469
the perimeter of CIRCLE C is: 56.5487

The distance between rectangle a and circle c is: 2.82843
CurveCut Name: CurveCut rc
X-coordinate: 6
Y-coordinate: 5
Radius: 9
Area: 56.3827
perimeter: 40.1372
the area of CurveCut rc is: 56.3827the perimeter of CurveCut rc is: 40.1372
The distance between rc and c is: 3Square Name: SQUARE - S
X-coordinate: 7
Y-coordinate: 9
side a: 12
Area: 144
perimeter: 48

the area of SQUARE - Sis: 144

the perimeter of SQUARE - S is: 48
Rectangle Name: RECTANGLE A
X-coordinate: 5
Y-coordinate: 7
side a: 400
side b: 300
Area: 120000
perimeter: 1400

the area of RECTANGLE Ais: 120000

the perimeter of SQUARE - S is: 1400
Shape Name: CIRCLE C
X-coordinate: 3
Y-coordinate: 5

the area of CIRCLE Cis: 254.469
the circumference of CIRCLE C is: 56.5487
Rectangle Name: CurveCut rc
X-coordinate: 6
Y-coordinate: 5
side a: 10
side b: 12
Area: 56.3827
perimeter: 40.1372

the area of CurveCut rc is: 56.3827

the perimeter of CurveCut rc is: 40.1372

Testing copy constructor in class CurveCut:
CurveCut Name: CurveCut rc
X-coordinate: 6
Y-coordinate: 5
Radius: 9
Area: 56.3827
perimeter: 40.1372

Testing assignment operator in class CurveCut:
CurveCut Name: CurveCut cc2
X-coordinate: 2
Y-coordinate: 5
Radius: 9
Area: 1136.38
perimeter: 220.137
CurveCut Name: CurveCut rc
X-coordinate: 6
Y-coordinate: 5
Radius: 9
Area: 56.3827
perimeter: 40.1372
```