CMPT125, Fall 2019

Homework Assignment 5
Due date: Monday, December 2, 2019, 23:59

You need to implement several classes in C++, and write a test for your solution. You may either implement everything in one file or use several files.

Submit everything in one **assignment5.zip** file to CourSys.

The assignment will be graded both **running** and **reading your code**.

Readability: Your code should be readable. Add comments wherever is necessary.
If needed, write helper functions to break the code into small, readable chunks.

Compilation: Your code MUST compile in CSIL using g++.
Use Lab 10 from CMPT 127 to see how to compile a cpp code.
Make sure that your code compiles without warnings/errors.

If the code does not compile in CSIL the grade on the assignment is 0 (zero).
Even if you can't solve a problem, make sure it compiles

Warnings: Warnings during compilation will reduce points.
More importantly, they indicate that something is probably wrong with the code.

Test! Test! Test! Test your code! This assignment does not contain a test file.
You need to write a test file and submit it as part of your solution.

Part 1 [20 points]

Reading material:
Copy constructor: https://www.geeksforgeeks.org/copy-constructor-in-cpp/
References (the & symbol): https://www.geeksforgeeks.org/references-in-c/

---

*Write a class **Point** representing a point in 2d.*
The class will have the following public methods:

```
// constructor - sets the coordinates to be x,y
Point(int x, int y)

// copy constructor - creates a new point with the same coordinates
Point(Point &copy)

// getters
int getX()
int getY()

// setters
void setX()
void setY()

// prints to screen the coordinates in the format (x,y)
void print()
```

Part 2 [20 points]

Reading material:
Abstract classes and pure virtual methods:
https://www.geeksforgeeks.org/pure-virtual-functions-and-abstract-classes/
---

*Write an abstract class **GeometricShape** representing an abstract geometric shape. The class will have the following public methods:*

```
// Constructor: gets a coordinate. The purpose of the
coordinate depends on the specific shape
GeometricShape(Point coord)


// returns the area of the shape
// returns 0 as default. To be implemented in each
concrete shape
virtual float getArea() { return 0; }

// returns the perimeter of the shape
// returns 0 as default. To be implemented in each
concrete shape
virtual float getPerimeter() { return 0; }

// virtual method. To be implemented in each concrete
method
virtual void print() = 0;
```

Part 3 [20 points]

Write two concrete classes implementing GeometricShape

*The class **Circle.** Circle will have the following public methods:*

```
// a constructor
Circle(Point center, int radius);

// returns the area of the circle
float getArea();

// returns the perimeter of the circle
float getPerimeter();

// prints the center of the circle and its radius
virtual void print();
```

*The class **Rectangle.** Rectangle will have the following public methods:*

```
// a constructor
Rectangle(Point topLeftPoint, int length, int width);

// returns the area of the rectangle
float getArea();

// returns the perimeter of the circle
float getPerimeter();

// prints the top left point of the rectangle, its length
and width
virtual void print();
```

Part 4 [20 points]

Reading material:
You may want to use the class Vector from STL
https://www.geeksforgeeks.org/vector-in-cpp-stl/
Also, you may want to use iterators
https://www.geeksforgeeks.org/iterators-c-stl/
For the countCircle problem, there are several solutions. One of them (not necessarily the best one) is to use dynamic_cast.
https://stackoverflow.com/questions/307765/how-do-i-check-if-an-objects-type-is-a-particular-subclass-in-c
---

*Write an abstract class **CollectionOfShapes**, representing a container of (pointers to) geometric shapes. The class will have the following public methods:*

```
// Default constructor
CollectionOfShape()

// adds the shape to the
void add(GeometricShape* shape);

// iterates through all the shapes and prints them
void print();

// returns the number of circles in the container
int countCircles();
```

Part 5 [20 points]
Write a function test() that tests your solutions.
Run your test() function from main().