Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2018
Homework # 4
Dynamic Memory and Name Spaces

Due date Nov 26th 2018

In this homework, you will extend your classes of HW2 with dynamic memory and name spaces.

You will define a new class for polygon. SVG element polygon defines a polygon that connects given 2D points. **Polygon** is a C++ class that represents an SVG element Polygon. The class defines a public inner class named **Point2D** to represent a 2 dimensional point.

Class **Polygon** also has at least the following functions:

- A constructor that takes an STL vector of Point2D objects to represent the **Polygon** points. There can be many 2D points in a **Polygon**.
- A conversion constructor that converts from a 2D Point.
- Three conversion constructors that convert from other **Rectangle, Triangle,** and **Circle** objects. Conversion from **Rectangle** and **Triangle** is trivial. For **Circle**, you should form a polygon by sampling the circle at 100 equal intervals and make a polygon.
- **Operator**[] that returns a reference to the point at the given index.
- **Operator**== that returns true if two **Polygons** are equal. Similarly operator != is also defined.
- **Operator**+ that adds two **Polygons** by concatenating the points and returns the result as a new object.
- **Operator<<** for producing the SVG code.
- Any other functions that need to be implemented with appropriate object oriented rules.

This class follow also the following rules.

- Use dynamic memory to keep your coefficients. Do not use STL vectors in your class data members.
- Make all error checks in your constructor and the member functions.

Your other new class **Polyline** is the same as **Polygon**, only it does not connect the first and the last points. **Polyline** uses **Polygon** for all the work other than the difference.

Your **Rectangle, Triangle,** and **Circle** classes the same as HW3.

Your **ComposedShape** class is very similar to your class in HW2. This time, this class will hold other shapes by converting them to polygons. Therefore, it has only a vector of Polygon objects in it. All the other functions and operator overloads are the same as HW2 and HW3.

You will also write a driver file that contains your main function. Your driver should make at least 5 objects Polygon class and ComposedShape class and should make array of objects of each class. Test each member function for each class. You should include the result SVG files from the draw functions in your submission.

Notes:

- Define and use your namespace.
- Use unnamed namespaces in one of your files.
- Do error and range checking for any parameters and user input.
- <u>Do not use any C++ features that we did not learn during the lectures.</u>
- For each class you will write its own header .h file and .cpp file for the separation of interface and implementation.
- As expected, you should follow all object-oriented programming principles.
- You should submit your work to the moodle page.
- You should submit the SVG files, the header and source code files and sample output results.