Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2018
Homework # 3
Operator Overloading
Due date Nov 10th 2018

In this homework, you will extend your classes of HW2 with operator overloading and other new capabilities.

Your **`Rectangle, Triangle,`** and **`Circle`** classes as very similar to classes in HW2 with the following additions.

- Each class should have 2 functions that return the perimeter length and the area of the shape.
- Each class should have overloaded **`operator<<`** for drawing(writing) the shape to an **`ostream`** object such as an SVG file or **`cout`** object.
- Each class should overload ++ and - - operators (both pre and post) for incrementing and decrementing the shape positions by 1.0.
- Each class should overload + and – operators that return a new shape object. These operators will accept a double as a parameter and they will add this double to the size of the shape to make the new shape.
- Operators == , !=, and other comparison operators to compare two shapes with respect to their areas.
- Each class should have static functions to return the total areas and perimeters of all shapes of their type created so far.

Your **`ComposedShape`** class is very similar to your class in HW2. It will have an inner public class **`ShapeElem`** that holds two variable members

- A **`void *`** that holds a pointer to one of the shape objects
- An enum variable that defines the type of shape object

`ShapeElem` class will have at least the following functions:

- Overloaded **`operator<<`** for drawing(writing) the shape to an **`ostream`** object such as an SVG file or **`cout`** object.
- Two functions that return the perimeter length and the area of the shape.
- Any other functions needed.

The class **`ComposedShape`** will use a vector of **`ShapeElem`** objects to keep the inner shapes. It will also have additional functions as follows

- Operator+= for adding a new shape to this composedShape
- Operator[] for returning a shape as an object of ShapeElem.
- **`operator<<`** for drawing(writing) the composed shape.

You will also write a driver file that contains your main function. Your driver should make at least 4 objects of each 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:

- Do error and range checking for any parameters and user input.
- Do not use any C++ features that we did not learn during the lectures.
- 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 will use all the object oriented techniques that we learned in the class including **`const, inline, declrtype, auto`** keywords.
- 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.