Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2018
Homework # 1
## Your First C++ Program
Due date Oct 19th 2018

You will write a C++ program that will optimize some 2D shapes and produce SVG (Scalable Vector Graphics) files to display calculated 2D graphics. SVG is a very easy XML language to learn and many web browsers directly support it. Below is a simple SVG file

```
<svg version="1.1"
     baseProfile="full"
     width="300" height="200"
     xmlns="http://www.w3.org/2000/svg">

  <rect width="100%" height="100%" fill="red" />
  <circle cx="150" cy="100" r="80" fill="green" />
  <text x="150" y="125" font-size="45" text-anchor="middle"
           fill="white">CSE 241</text>
  </svg>
```

and the resulting graphics is below.
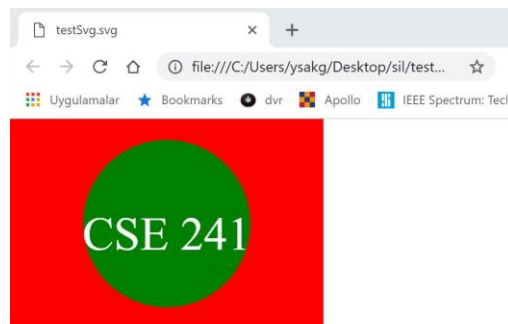
You may learn basic SVG syntax and objects from many available sources on the web. One good source is **https://developer.mozilla.org/en-US/docs/Web/SVG**.

Your program will have a main container shape (rectangle, circle or triangle). It will try to fit many small shapes (rectangle, circle or triangle) in the main container shape by carefully placing these shapes into the container shape.

Below is a sample output of your program.

```
Please enter the main container shape (R, C, T)
R
Please enter the width
200
Please enter the height
100
```
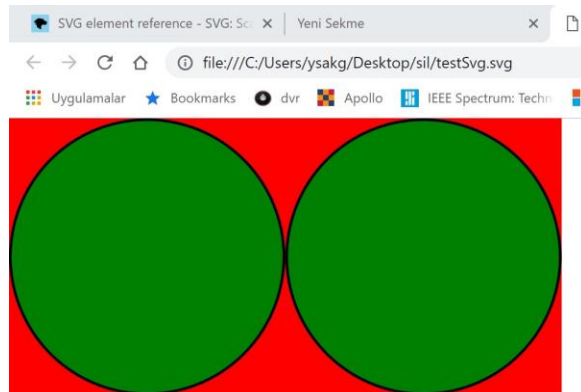
```
Please enter the small shape (R, C, T)
C
Please enter the radius
50
I can fit at most 2 small shapes into the main container. The empty area (red) in
container is 12146.01836.
```

The result file should be stored in result.svg. For the above input, it should look like

Note that, your program should try to fit as many small shapes as possible into the main container. Small shapes should not overlap. Your triangles can be only equilateral triangles. Your rectangles and triangles can rotate to fit the main shape better.

Notes:
  • Do not use any functions from the standard C library (like **printf**)
  • You will use C++ file streams to write to your SVG files. You will define C++11 Enum classes for the shapes.
  • Your program should have only functions and no classes other than enums.
  • Do not forget to indent your code and provide comments.
  • Check the validity of the user input.
  • **Test your programs very carefully at least with 10 different runs and submit your result.svg files for each. There should be cases for unexpected values such as too small containers. Your program should be compiled and run on the VM we provide.**
  • You should submit your work to the moodle page.