**Assignment 3**

The assignment is graphics-drawing program (see the screen capture below). Here is how the program is supposed to work: clicking a button labeled “circle”, “rectangle”, or “triangle” allows for drawing a *fixed-size* shape. Then, one clicks the mouse on the canvas area a shape to draw an indicated shape (circle, rectangle, or triangle) at the mouse location. Note that all previously drawn shapes are always visible.

**Requirements:**

1. Design and implement the program as specified above.
2. A class diagram that shows the relations of the classes and/or interfaces used in the program.
3. A sequence diagram that captures the sequence of method calls when mouse is clicked.
4. Briefly explain your design consideration: Was there a design goal you wanted to accomplish? What was the consideration for creating the classes and interfaces as you did? Were there ways that your program could have been simpler? In what aspect, if any, was your design flexible?

**Some design hints:**

1. In the future, the program may need to be modified to add additional buttons for more kinds of shape, or remove some existing buttons. So you want to design the program in such a way that when such changes happen, they happen in some isolated places that you can easily make some changes in code. In other words, you want to keep the existing code structure intact as much as possible. For instance, if you do use the method *paintComponent()*, you want to make it unaware of changes in shape kinds. To put the idea in code, g.*draw(shapeObject)* would be drawing any kind of shape, noting that in Java, all concrete shapes (Polygon, Rectangle, etc.) are subclasses of Shape (an abstract Java class). To make such a design possible, we typically write a “factory” method to create a shape object whenever mouse is clicked.
2. There are different ways to design this program. For example, you can also do it without using *paintComponent()* method. In any case, here is a sequence events:
3. When a button is clicked, information about the shape is saved somewhere for later retrieval.
4. When a mouse is clicked, retrieve the shape info, and create a new shape object (possibly by a shape factory). Then, draw the object at the mouse location.
5. Note that if you call “*repaint()*” method (which would in turn calls *paintComponent()*), then it would erase the current drawing on the canvas, and paintComponent() method would put new drawing on. Thus, as you create new shape objects, you save them too for later use.

That being said, you **feel free** to make your own design decisions, and describe it in above Requirement 3.

