



---

## Lab7: Mini-Paint Application

### Objectives

Upon completion of this assignment, you will be able to:

- Design an object-oriented model for geometric shapes
- Apply the OOP concepts of inheritance and polymorphism to your design.
- Create an advanced GUI with 2D Graphics capabilities

### Part 1: Geometric Shapes Data Model

#### Tasks:

1. Design an object-oriented model that covers the following geometric shapes: Line Segment, Circle, Rectangle.
2. Apply the concepts of inheritance and polymorphism to your design.
3. Stick to the following interfaces in your design:

```
public interface Shape{  
    /* set position */  
    public void setPosition(java.awt.Point position);  
    public java.awt.Point getPosition();  
    /* colorize */  
    public void setColor(java.awt.Color color);  
    public java.awt.Color getColor();  
    public void setFillColor(java.awt.Color color);  
    public java.awt.Color getFillColor();  
    /* redraw the shape on the canvas */  
    public void draw(java.awt.Graphics canvas);  
}
```



```
public interface DrawingEngine {  
    /* manage shapes objects */  
    public void addShape(Shape shape);  
    public void removeShape(Shape shape);  
    /* return the created shapes objects */  
    public Shape[] getShapes();  
    /* redraw all shapes on the canvas */  
    public void refresh(java.awt.Graphics canvas);  
}
```

## Part 2: Drawing and Painting Application

### Description:

Drawing and painting applications are very popular and have a huge user base.

They generally offer a big number of features that includes but is not limited to: Drawing, Coloring, and Resizing. They also include a number of built in, and possibly extensible sets of geometric shapes, and classically, they allow the user to undo or redo any instructions so as to make the application more usable.

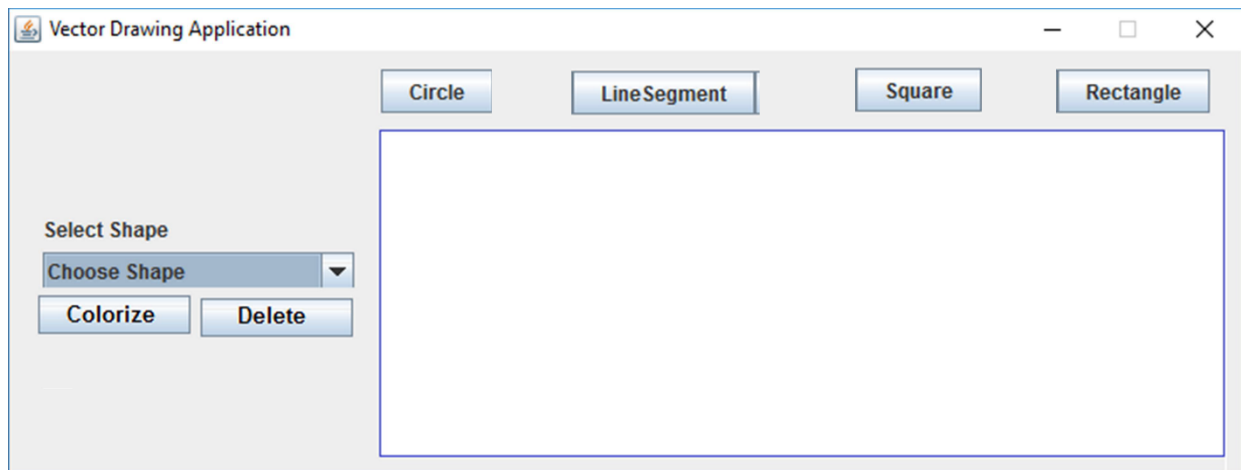
In this lab, we will start with the basic feature for a paint application, then we will extend it in the next lab.

### Tasks:

1. Implement your design from part 1 in Java.
2. Design and implement a GUI that allows the drawing functionalities for the user on all the shapes defined in part 1.



3. There is no need to use the cursor in drawing. Just use a set of buttons to select the shape. Upon pressing the button, a dialog box appears to enter the properties of the shape such as: position, radius, length... etc.



4. In order to support the operations of add, colorize and delete a shape, there should be a way to select the shape. Add a drop-down menu that shows all the drawn shapes. You can use any arbitrary names for the drawn shapes (For example: circle01, square\_04... etc.). That way, the user can select a shape then apply an operation on it.
5. You should use Swing graphics library

### Part 3: UML

- You are required to deliver the UML diagrams (Class diagram and Use case diagram ) for question 2 and 3 in lab 3.
- **reminder:** Question 2 that talks about the small library system and question 3 talks about the company and the institution.
- You should use any online tool for drawing the UMLs. Don't use your pencil to draw.



---

### What to be delivered

- On the Google form, you should deliver a zipped file that contains the .java files of your classes along with your UMLs.
- Your zip file should be named as id\_groupNumber. For example, 4678\_G2.

**Note: An online discussion will be made with you on what you have delivered.**

### Policies

- You should work individually.
- Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.
- No late submission is allowed