# COMP10062: Assignment 8

## The Assignment

In this assignment you will create a very simple drawing app. The focus is on proper exception handling and the use of mouse listeners.

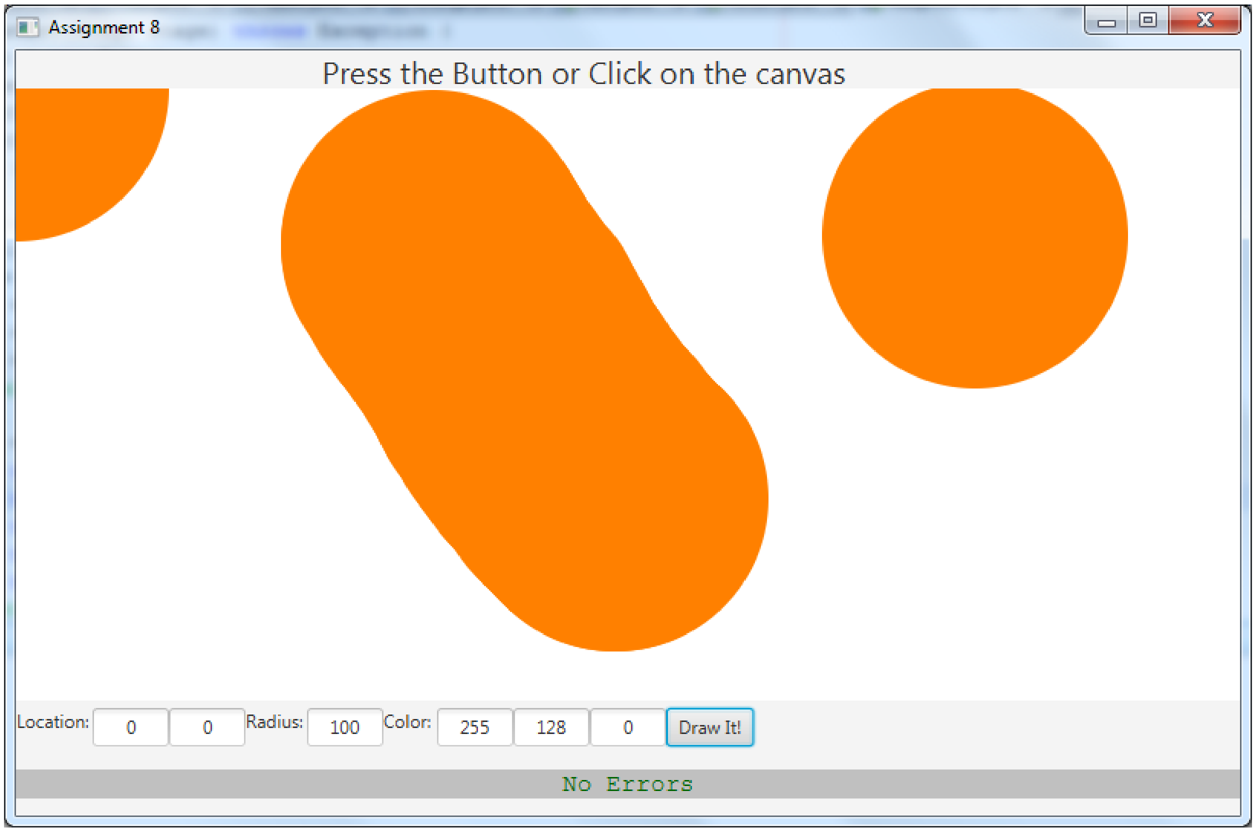
The app you create must be capable of drawing one kind of shape (circle, rectangle, oval, etc.). The user must be able to configure the parameters of the shape and must be able to draw it by pressing a button, by clicking the mouse on the canvas, and by dragging the mouse across the canvas.

## Performance: Basics

The screen capture below shows a very bare-bones version of the app. It shows one circle that was created by pressing the "Draw It" button, another that was created by clicking, and a trail of circles created by dragging.

Your app can draw circles like the one shown below, or you can choose a different type of shape.

Your app should not look exactly the same as the one below. You should try to put some thought into what would make an attractive and usable design.



The user must be able to configure the color, size and location of the shape. The location fields will only be used when the "Draw" It button is pressed. Otherwise, it uses the mouse location.

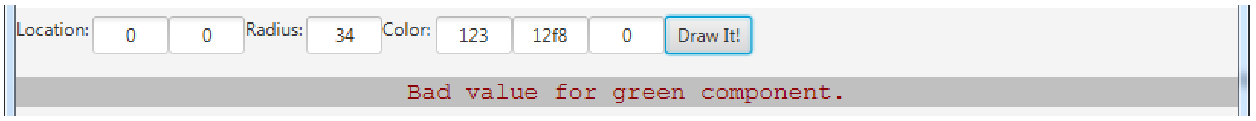
## Performance: Error Handling

It should be impossible for the user to trigger an exception. When the user clicks “Draw It” or clicks or drags the mouse, any possible exceptions (badly formatted fields, out of range color values, etc.) should be caught and the user should get an error message in a label that specifies:

1. The name of the field that caused the error.
2. The nature of the error (out of range vs. non-numeric)

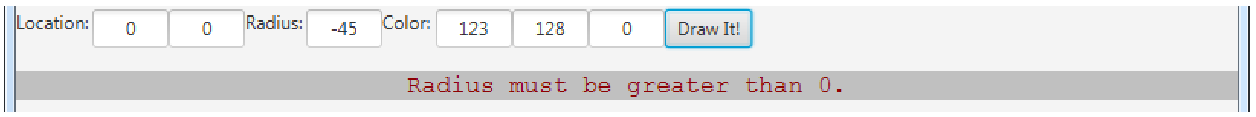
If more than one field causes an error, you only have to describe one of them in the label.

Here are some examples:



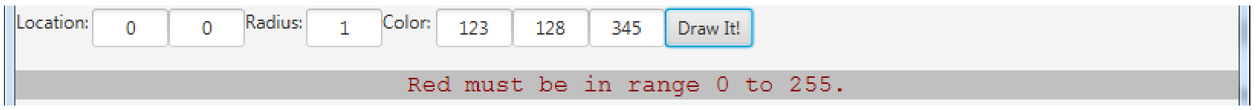
Location: 0 0 Radius: 34 Color: 123 12f8 0

```
Bad value for green component.
```



Location: 0 0 Radius: -45 Color: 123 128 0

```
Radius must be greater than 0.
```



Location: 0 0 Radius: 1 Color: 123 128 345

```
Red must be in range 0 to 255.
```

When the user successfully draws a shape, the error label should be reset to “No Errors” or something similar. The color of the label should change to indicate errors vs. no errors.

## Optional Extras

There is so much more that you could do with this! All of the ideas below are optional.

- Allow a variety of shapes
- Set colors using a color chooser or some other method
- Set the location fields with a mouse click
- Clear the screen
- Highlight error fields by changing their colors
- Add effects to the drawing (gradients, reflections, etc.)
- Maintain an array of shape objects so that the user can delete or modify similar objects
- Etc…

## Handing In

**You have approximately 1 week to complete this assignment**. See the due date and time on the drop box. Hand in by attaching a zipped version of your **.java** (not .class) files to the drop box.

Make sure you follow the **Documentation Standards** for the course.

## Evaluation

Your assignment will be evaluated for performance (40%), structure (40%), and documentation (20%) using the rubric in the drop box.