

# CSCI 202: Object-Oriented Programming

## Lab 2 - Due Thursday February 15, 2024 at 11:59pm

### Resources:

- Java API index: <https://docs.oracle.com/en/java/javase/17/docs/api/index-files/index-1.html>

### Drawing a House in JavaFX

JavaFX is a framework for creating graphical user interfaces (GUIs) in Java. In this lab you will be tasked with creating a JavaFX program that draws a house. When your program is finished it will look similar to the follows image below (see Figure 1):

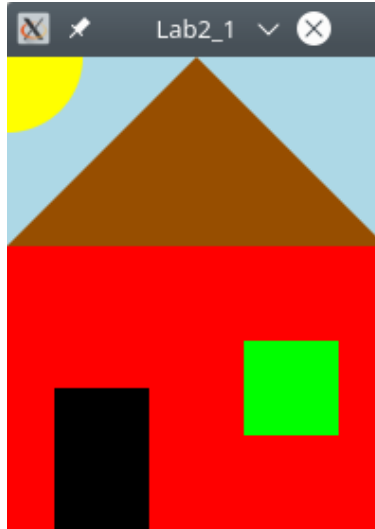


FIGURE 1. Completed Program

The origin (point  $(0,0)$ ) of the coordinate system in JavaFX is in the upper left corner. The  $x$ -axis moves horizontally from left to right, and the  $y$ -axis moves vertically from top to bottom (see Figure 2). The units in the coordinate system are measured in pixels (using type `double`).

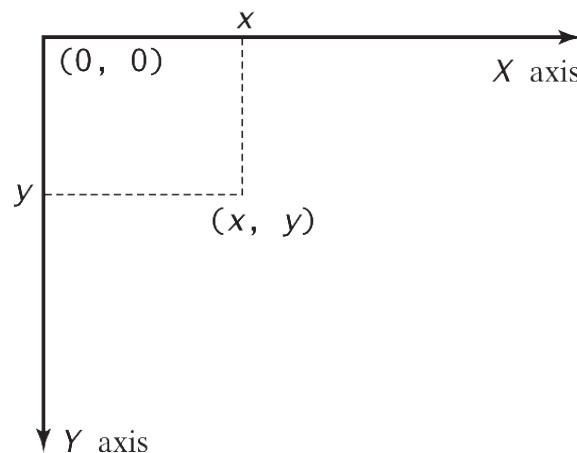


FIGURE 2. JavaFX Coordinate System

The house will be drawn by creating several shapes. These shapes will each be added to a **Group** object. A **Group** is a JavaFX component that contains a list of children nodes (a node is a graphical component in JavaFX). Each children node in a **Group** is rendered (or drawn) to the screen in the order they are added. In case of overlap, nodes added later to a **Group** are drawn over ones added earlier.

1. Download and unzip [Lab2\\_1.zip](#) onto your local computer. Then open up the unzipped project with IntelliJ.
2. In IntelliJ open up Lab2\_1.java (this is in package lab2\_1).
3. In the `start` method create a new Group by writing:

```
Group group = new Group();
```

4. Next, create a rectangle for the red walls of the house by writing:

```
Rectangle walls = new Rectangle(0, 100, 200, 150);
walls.setFill(Color.RED);
group.getChildren().add(walls);
```

The constructor `Rectangle(double x, double y, double width, double height)` creates a new `Rectangle` object with an upper left corner at position  $(x, y)$  and with given width and height.

The `setFill(Color c)` method sets the interior color of the rectangle.

If we wanted to set the color of the lines drawn around the rectangle we could have call the rectangle's `setStroke(Color c)` method.

The last line simply adds the rectangle to our group.

5. Add a door and a window to the house by creating rectangles similar to the previous step.

To get the color black we can use `Color.BLACK`.

We can create a custom color by using the constructor `Color(double red, double green, double blue, double opacity)`.

The red, green, blue, and opacity components of a color is specified as a double from 0.0 (completely off) to 1.0 (completely on). For Example, to create an new `Color` object that is green we can write:

```
new Color(0, 1, 0, 1)
```

6. The roof can be drawn as a triangle by creating a `Polygon` object:

```
Polygon roof = new Polygon();
roof.setFill(Color.web("#974E00")); // roof color is brown
```

The `Color.web(String colorString)` method creates a color specified with an HTML or CSS attribute string. See [http://docs.oracle.com/javafx/2/api/javafx/scene/paint/Color.html#web\(java.lang.String\)](http://docs.oracle.com/javafx/2/api/javafx/scene/paint/Color.html#web(java.lang.String)) for more information.

7. Now add three points to roof:

```
roof.getPoints().addAll(0.0, 100.0); // lower left point
roof.getPoints().addAll(100.0, 0.0); // top point
roof.getPoints().addAll(200.0, 100.0); // lower right point
```

8. Add roof to our group:

```
group.getChildren().add(roof);
```

9. Draw the sun by creating a circle:

```
Circle sun = new Circle(0, 0, 40); // center (0,0), radius 40
sun.setFill(Color.YELLOW);
```

10. Now create a pane and add our group to it:

```
Pane mainPane = new Pane(group);
```

11. Set the background color of `mainPane` to lightblue:

```
mainPane.setStyle("-fx-background-color: lightblue;");
```

The `setStyle(String value)` method is used to set the style of a node using CSS.

12. Finally, create a scene that is 200 pixels wide and 250 pixels high and place it on the main stage:

```
Scene scene = new Scene(mainPane, 200, 250);
stage.setTitle("Lab2_1"); // Set the stage title
stage.setScene(scene); // Place the scene in the stage
stage.setResizable(false); // Make window non-resizable
```

```
stage.show();           // Display the stage
```

When you are done, zip up your project in a file called [Lab2\\_1.zip](#) and upload this .zip file to Canvas (click on [Assignments](#) and go to [Lab 2](#)).