Lab 9  Name:______________________________  Checked:______

Objectives:
Introduction to graphics and JavaFX.

Preparation: Get to know the Java coordinate system
Download Snowman.java. Compile and run it to see the image it produces.

1. Note that in every GUI-based JavaFX class, certain imports must be used as indicated in the listing. How does the program get access to the classes in “Application, Scene, and Stage? Find the specific import statements and copy them here:

   ____________________________________________  ____________________________________________

2. Looking at the code for Snowman.java, how is the title (“Snowman”) of the window specified? Copy the exact code:

   ____________________________________________

3. How is the background color of the window specified?
   Copy the exact code:

   ____________________________________________

4. The line below specifies a circle representing the snowman’s head. What is the name of the class? What is the name of the object? What are the (x,y) coordinates and radius of the circle?

   Circle head = new Circle(80, 70, 30);

   class:___________________  object:____________________ (x,y)=(________,________) radius = _______

5. Using the first grid on the next page (Sketch 1), draw the circle representing the head. Be careful to place it in the correct position, with the correct radius.

6. Refer to the remaining code to complete the sketch of the snowman in the coordinate system. Be sure to draw and position precisely in the grid all the shapes corresponding to the three white blobs representing the snowman plus the lines representing the arms. The rest (eyes, hat, buttons, sun, ground) is optional.

7. Note that the snowman is defined in its original position and is then repositioned using the code:

   snowman.setTranslateX(170);
   snowman.setTranslateY(50);

   For Sketch 2 draw the snowman in its position after repositioning this time 1) include the sun and ground; and 2) label the centers of all the circles and ellipses with their coordinates.

8. For this lab, we will create a new application called FancySnowman.java by modifying the current Snowman application. Begin by saving the current code for Snowman.java under the new name. Remember to change the name in all places where it occurs, including the comments. Verify that it still compiles and runs (it should still look the same). Next, change some of the colors – refer to the JavaFX Color class for inspiration: https://docs.oracle.com/javase/8/javafx/api/javafx/scene/paint/Color.html
   • Change the color of the ground to gray
   • Change the color of the sky to pink
   • Change the snowman's eyecolor to blue.

Scan or take a picture of your sketches on the next page.
Submit (1) sketches and (2) FancySnowman.java through blackboard under “Lab 9 Prep”. 
Sketch 1: BEFORE REPOSITIONING

Sketch 2: AFTER REPOSITIONING INCLUDING SUN and GROUND
Part A
Check the work you did for preparation with your partner.

- Answers to all the questions
- Sketch 1
- Sketch 2
- Labels for at least 5 points
- FancySnowman.java

Classmate signature: ________________________________

Part B
Proceed step by step with the following changes. For each one, try first sketching it on Sketch 1 (from the preparation), to help work out the correct coordinates. Then, write the code to implement it and test it.

1. Change his arms so that the snowman is waving with the right arm instead of the left one.

2. Add a highlight to the snowman's hat. Be sure to add the highlight to the hat Group.

3. Change the sun so that it is larger and is positioned in the top left corner of the window (see final image below)

4. Change the order of the shapes comprising the snowman group, so that one of the arms is behind the snowman

5. Create five or more rays of sunshine (i.e., yellow lines) for the sun. Hint: you can change the color of a Line using setStroke(Color.YELLOW); Then, rather than adding the rays individually to the root Group, create a new Group comprising the rays and then add the rays Group to the root Group of the scene.

   - Remember to note of the rays in Sketch 1. When done, add the rays Group to the group hierarchy on the chart shown in Part C of this handout.

6. Move the snowman further to the right by 30 pixels (be sure to do this using the translate method)

7. Add an arc to replace the mouth of the snowman, using the following code:

   ```java
   Arc mouth = new Arc(80, 70, 20, 15, 220,100);
mouth.setType(ArcType.OPEN);
mouth.setStroke(Color.BLACK);
mouth.setFill(null);
   ```

   This should make the snowman smile!

8. Optional: Make him look like he is winking!
9. Add a Text object to display a message of your choice. Refer to the examples Einstein.java or HelloJavaFX.java for a refresher on how to do that, including how to import the Text class.

10. Create a custom color using the code:

```
Color dirtySnow = Color.rgb(200,220,220);
```

Now use this color for the ground:

```
ground.setFill(dirtySnow);
```

11. Create a random color for the sky, by supplying values in the range 0-255 for each of the red, green, and blue components. (Remember to import the Random class.)

12. Add an extra button to the snowman’s torso, but rather than listing it along with the other elements in the snowman Group, add it after the snowman Group is defined, using the following code:

```
Circle extraButton = new Circle(80, 160, 6);
extraButton.setFill(Color.GREEN);
snowman.getChildren().add(extraButton);
```

13. Using a similar approach, add an extra ray to the rays Group after that group is defined.

14. Create a new Group named snow, initially empty and include this group in the root group. Using a loop, add “snowflakes” to the snow group, i.e., small white circles of random size and in random positions throughout the window (this is similar to adding the extra button, above). **Hint:** You did something similar with buttons and rays. The snowflakes need to be added using a loop, but each one is very similar to adding an extraButton to the snow Group.
Part C
Create a new group to organize the snowman head, eyes, mouth and hat so that they can be moved together. Redraw the group hierarchy to reflect this change. Test that it still works before moving it (it should look the same) and then use this new group to reposition and tilt the head to the left.

group hierarchy updated:
Lab 9 Comments  Name: ___________________  Checked: ______

Comments on this lab, please:

What was the most valuable thing you learned in this lab?

What did you like best about this lab?

Was there any particular problem?

Do you have any suggestions for improving this lab as an effective learning experience?