Project 2: Shapes Drawings/Images

Victoria Lee

UMGC: University of Maryland Global Campus

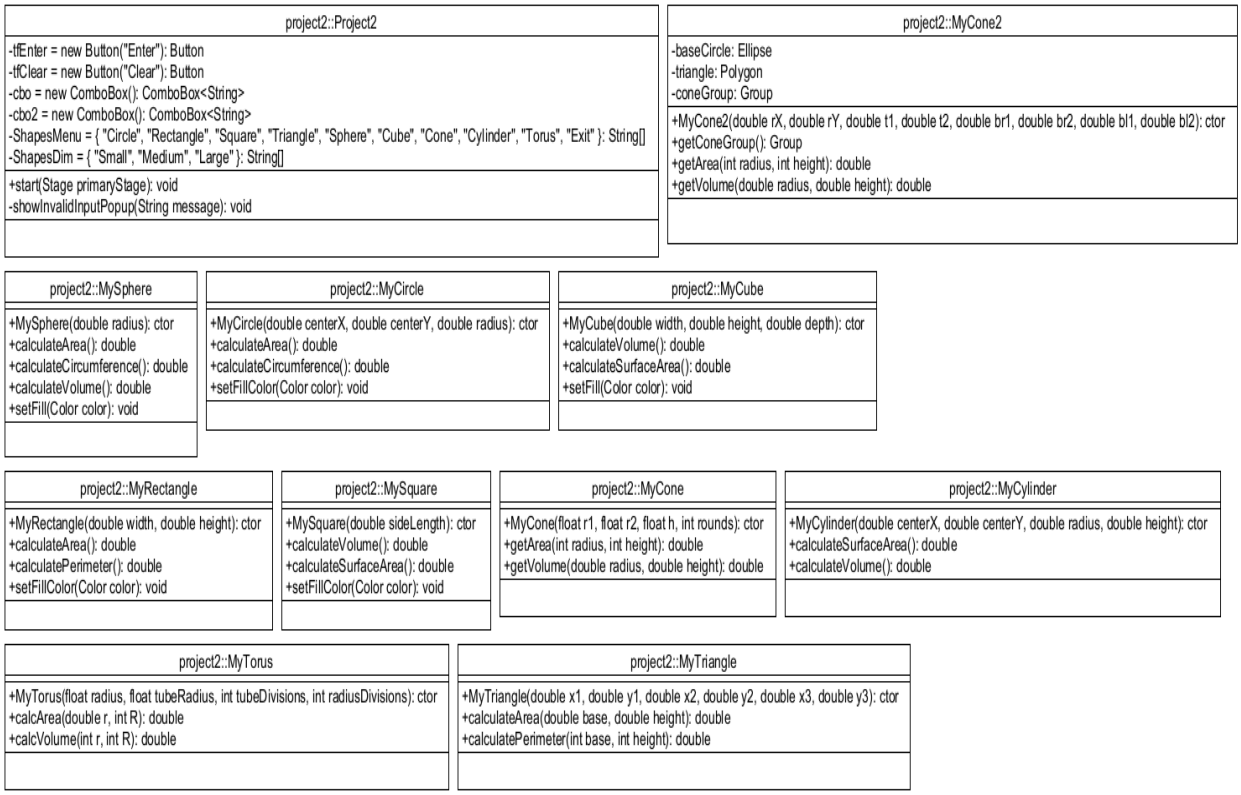
CMSC 335: Object-Oriented and Concurrent Programming

9/4/2024

UML Diagrams:

Note: The Main Method is in Project1 Class. This allows it to run the program smoothly. The package is project1. The UML diagrams below include the package.

UML Class Diagrams and Package:

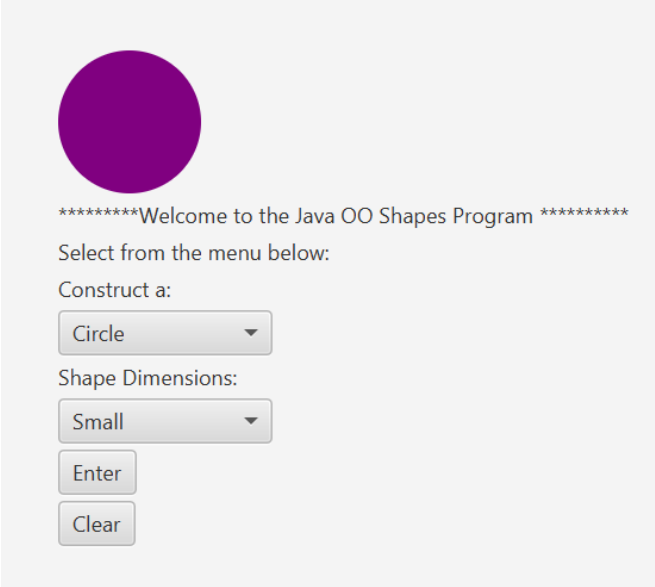


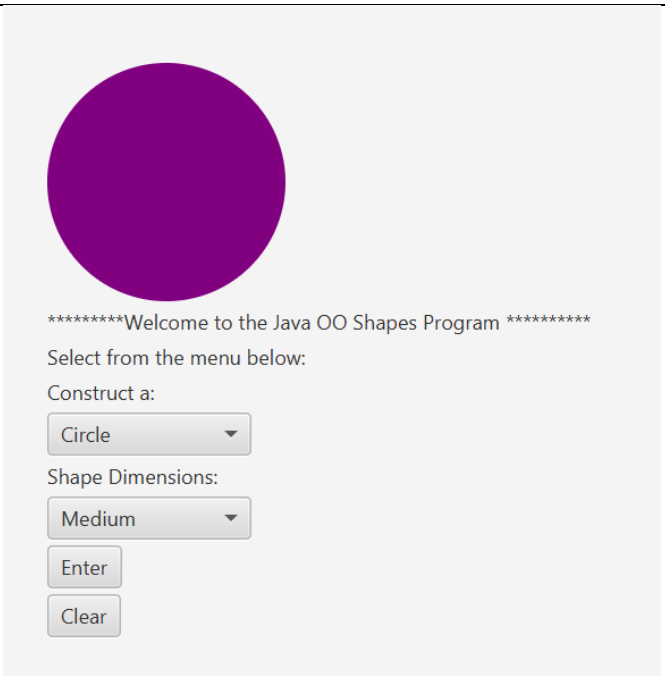
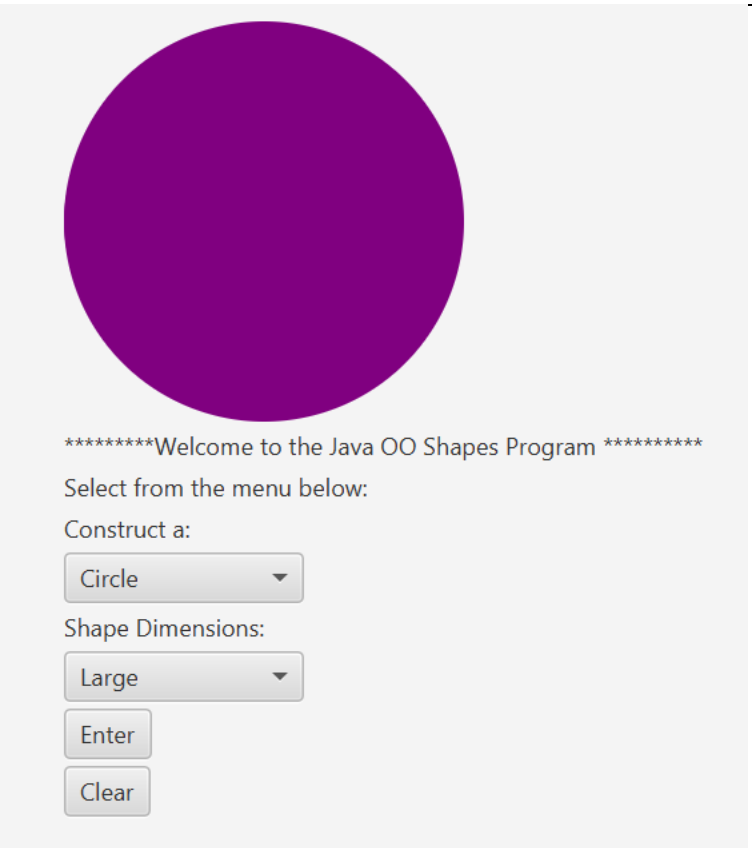
Developer's Guide, Test Cases, and Lessons Learned:

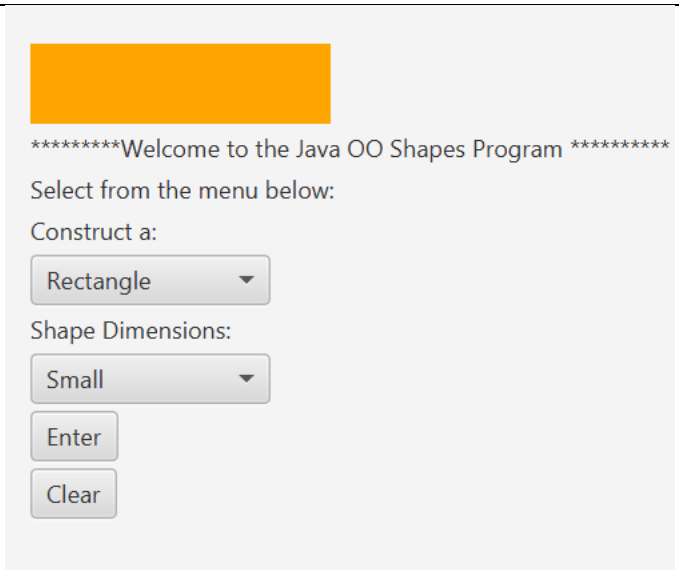
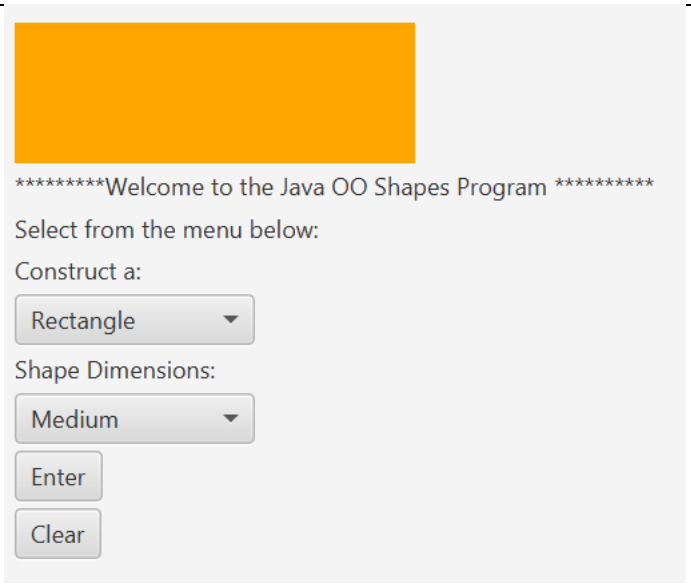
You can import the files using the new project and import all of the classes. Make sure to have a package called project1 in the "src" file. An alternative way is to create the project and import all the files including the project1 package I already included from the ".ZIP" file. You can compile the file and execute the program by going to the Project 2 class, right-clicking on the class, and selecting the run option. Make sure it is allowing the JavaFX to run. It should pop up the Shapes Program in the console and then you can choose the options below.

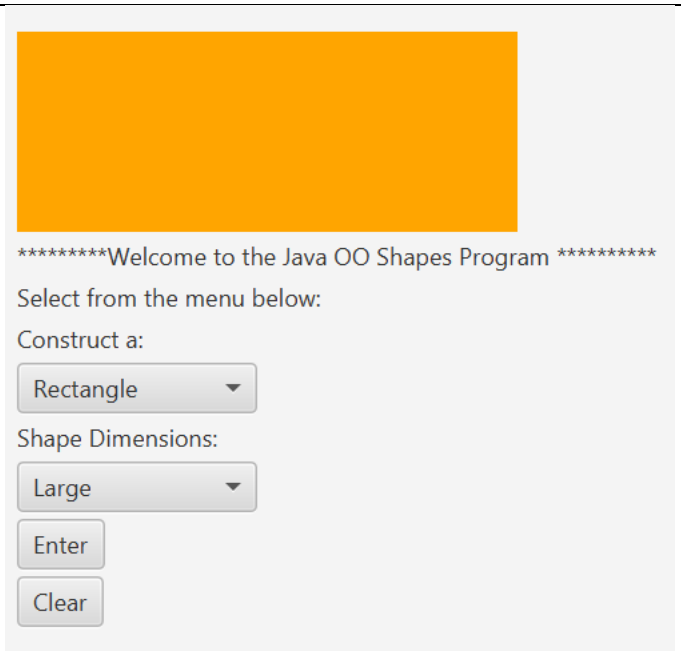
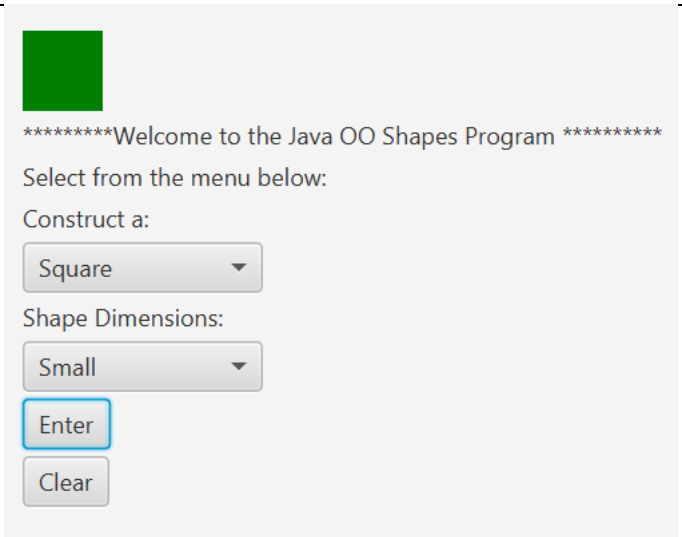
Note: After entering and drawing or showing the image of the shape, you must press the clear button continuously until it is a blank background by default. And then you can go ahead and select another shape to draw. I made it this way if I were to put the code pane.getChildren.clear(), it would clear the whole program, so this was how I made it. Also, if two of the same shape appear, it is because I wasn't sure which one would get credit more, so I put both (drawing and image) together.

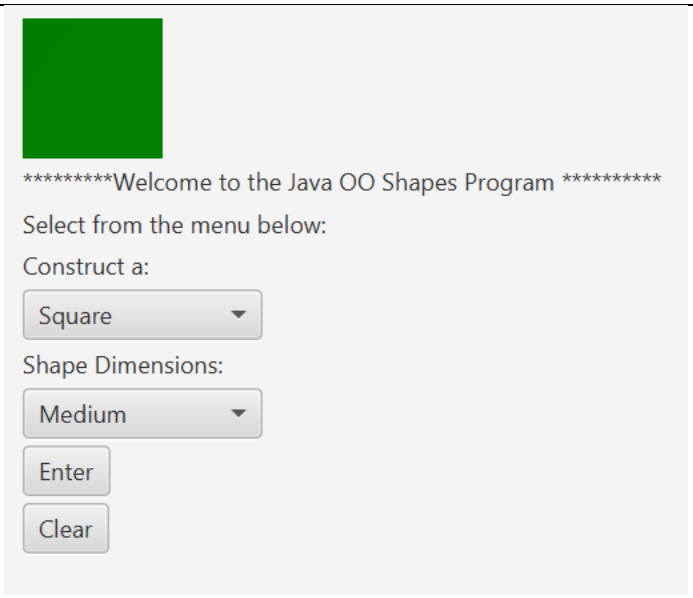
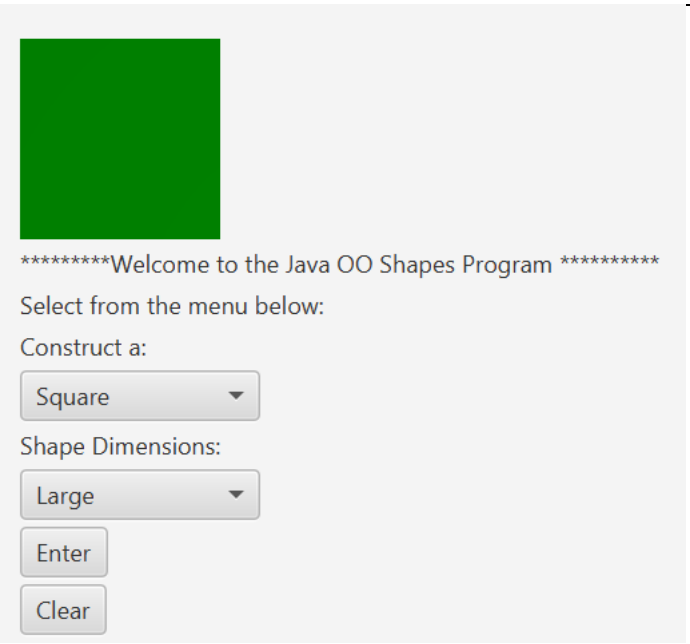
Table 1 (below): *Developer's guide describing compiling and executing the program. Documentation includes Lessons learned at the end.*

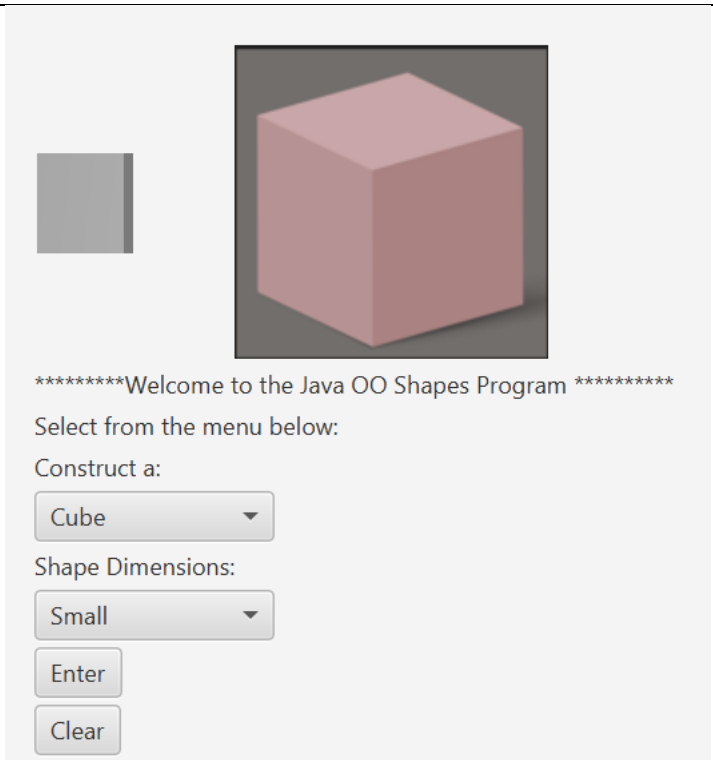
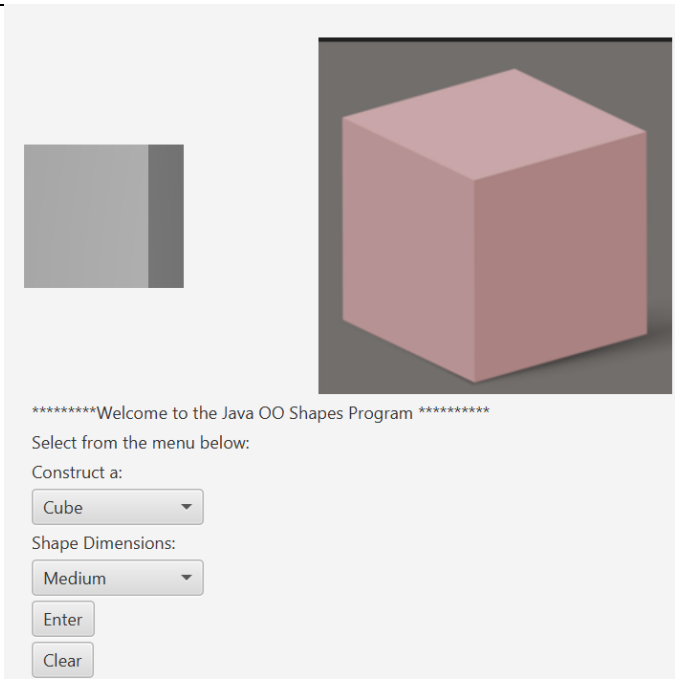
Test #	Description	Screenshot	PASS / FAIL Flag
1.	To test the Small Circle. Select Circle Select: Small Then Press Enter Then Press Clear after done until blank.		PASS

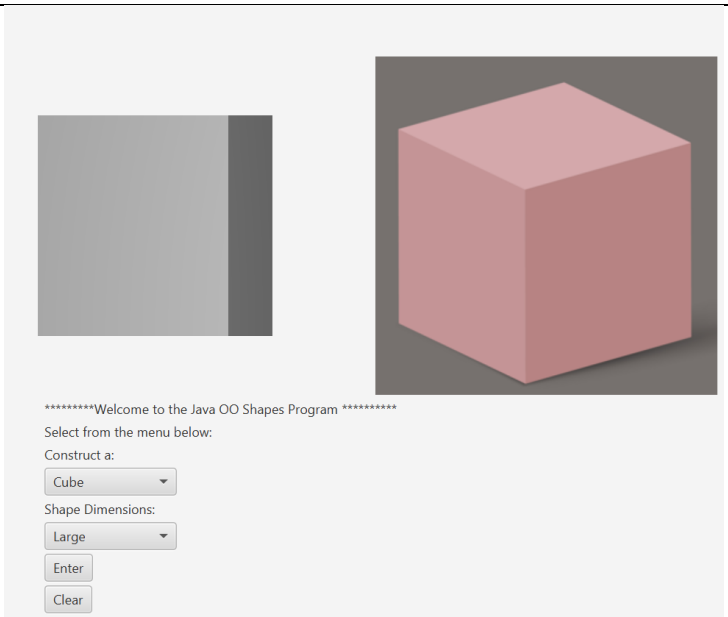
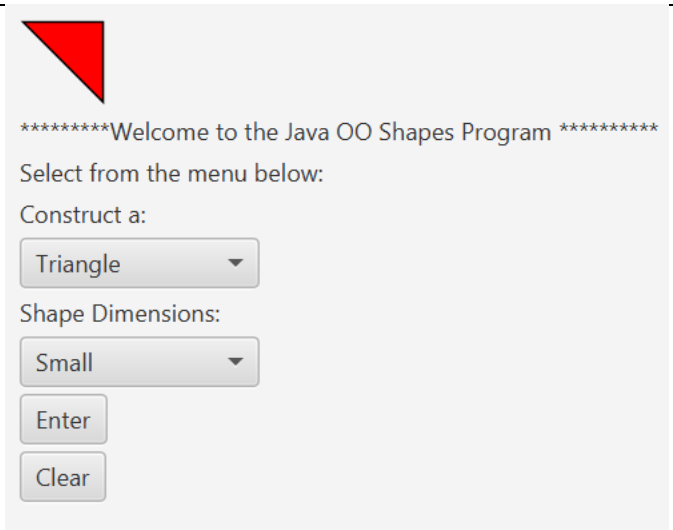
2.	To test the Medium Circle. Select: Circle Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS
3.	To test the Large Circle. Select Circle Select: Large Then Press Enter Then Press Clear after done until blank.		PASS

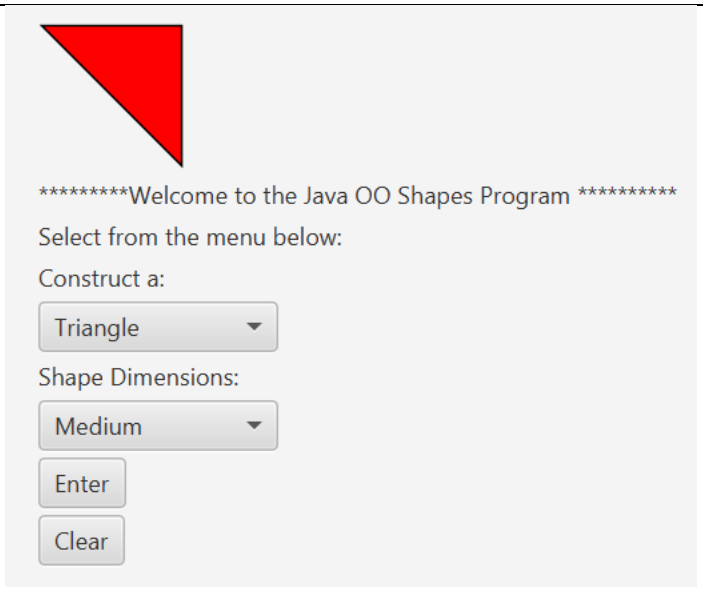
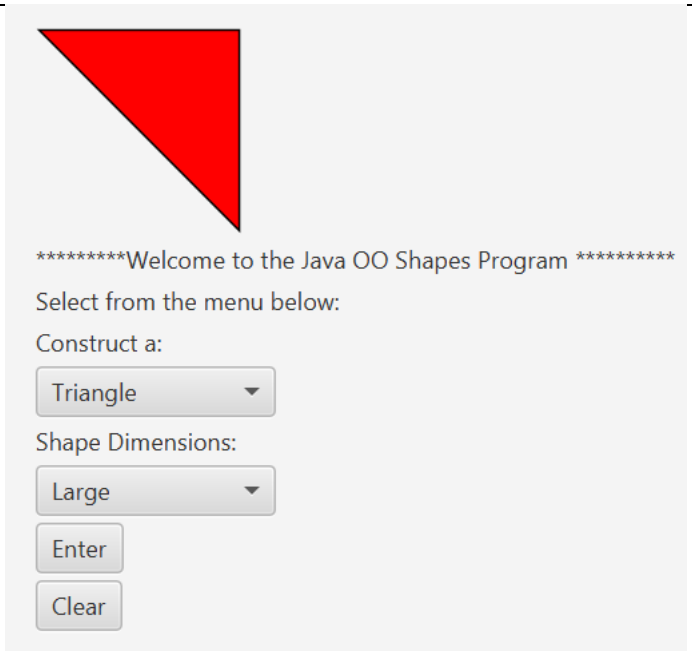
4.	To test the Small Rectangle. Select: Rectangle Select: Small Then Press Enter Then Press Clear after done until blank.		PASS
5.	To test the Medium Rectangle. Select: Rectangle Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS

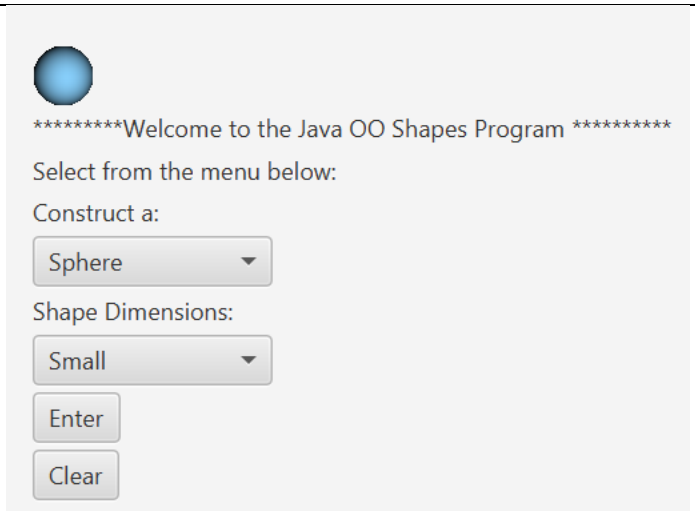
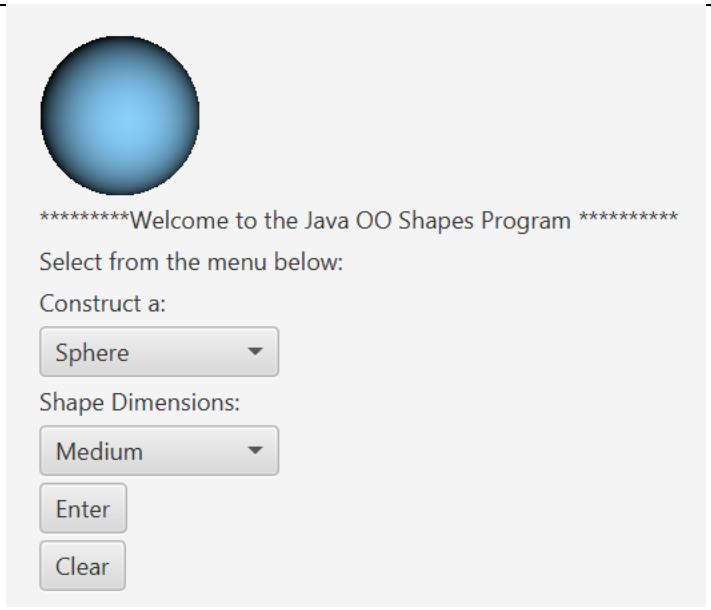
6.	To test the Large Rectangle. Select: Rectangle Select: Large Then Press Enter Then Press Clear after done until blank.		PASS
7.	To test the Small Square. Select Square Select: Small Then Press Enter Then Press Clear after done until blank. Then Press Enter Then Press Clear after done until blank.		PASS

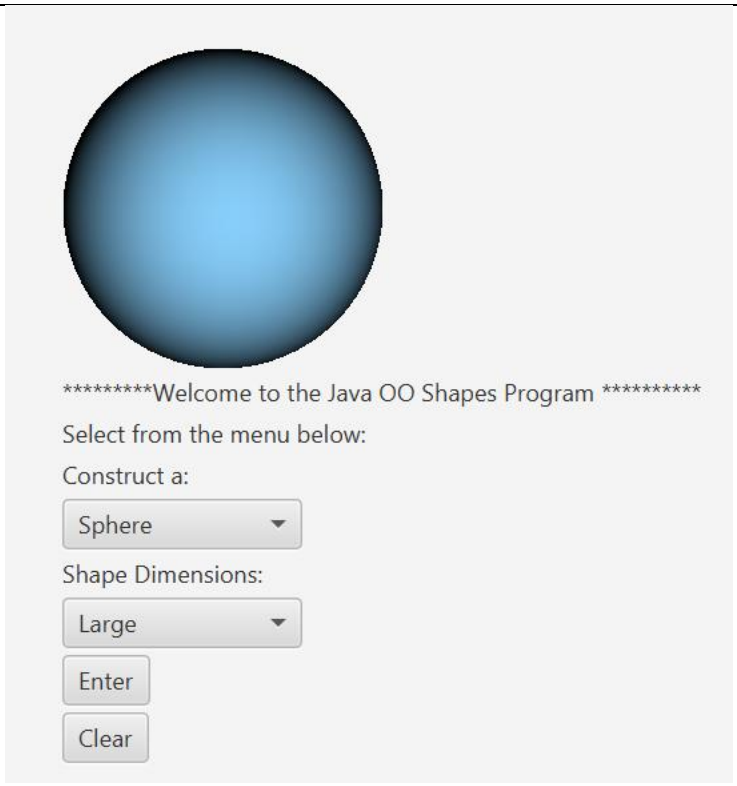
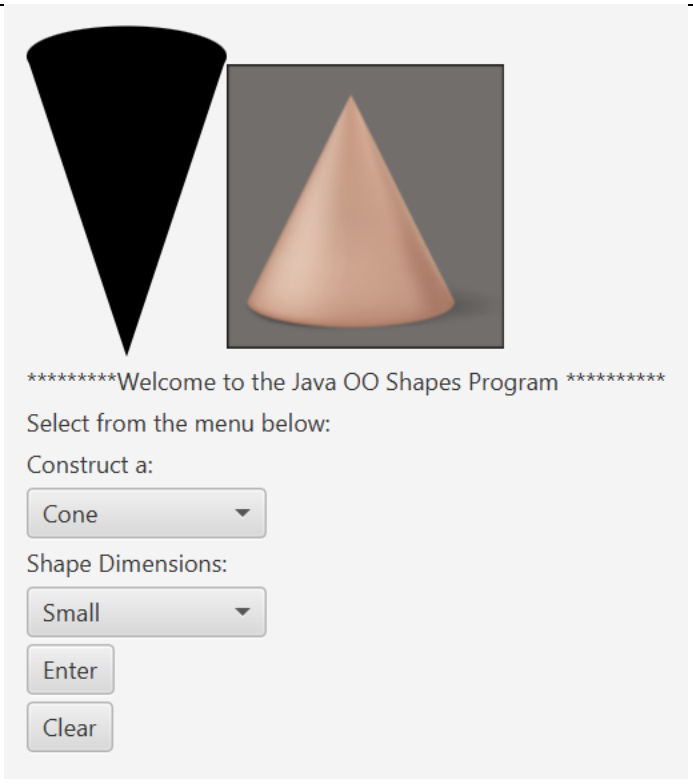
8.	To test the Medium Square. Select Square Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS
9.	To test the Large Square. Select Square Select: Large Then Press Enter Then Press Clear after done until blank.		PASS

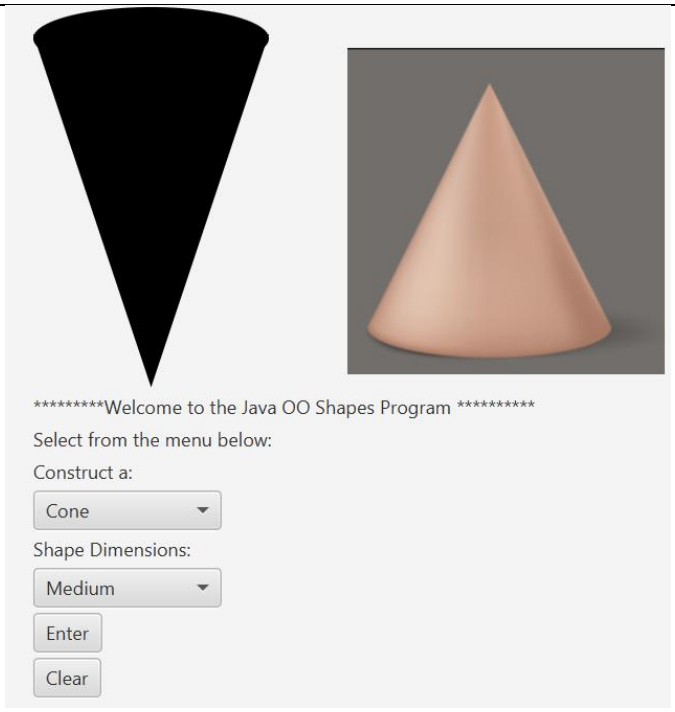
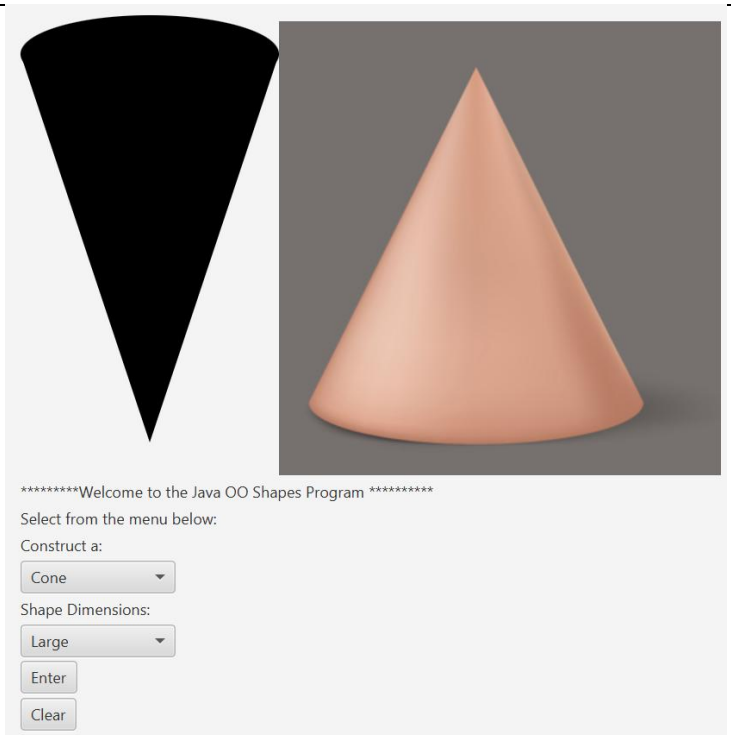
10.	To test the Small Cube. Select: Cube Select: Small Then Press Enter Then Press Clear after done until blank.		PASS
11.	To test the Medium Cube. Select: Cube Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS

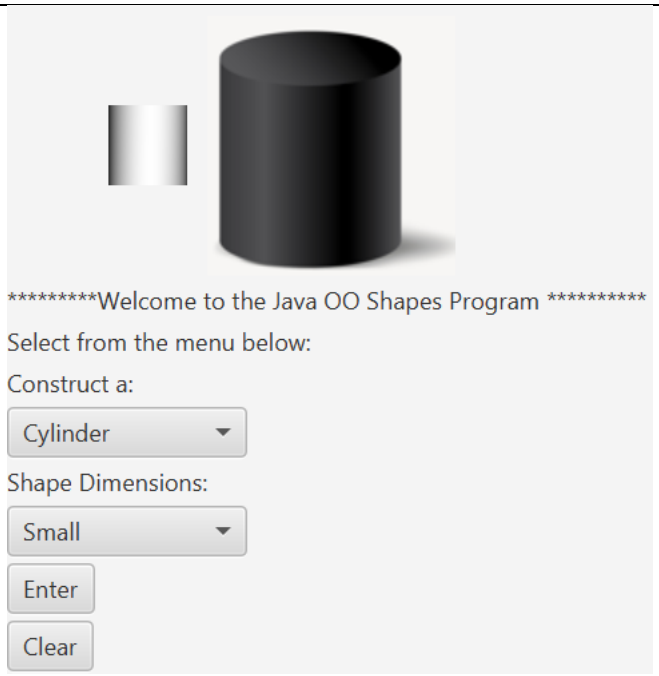
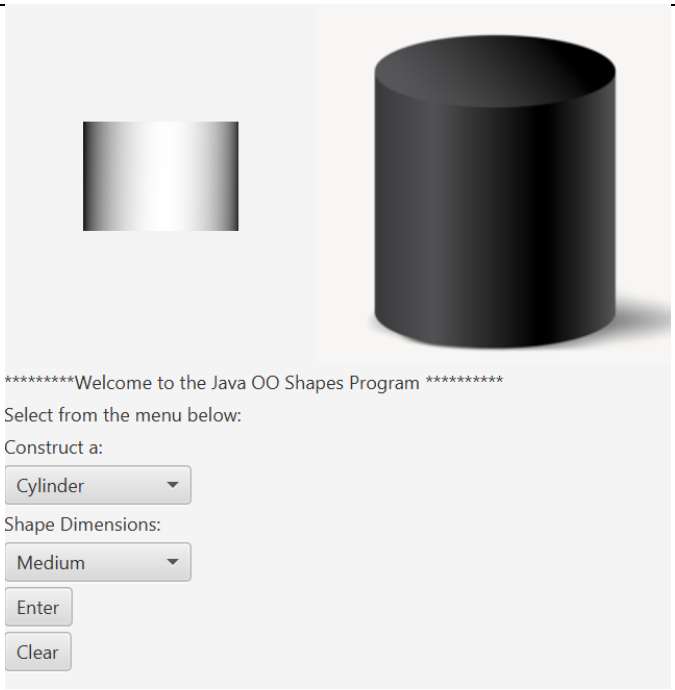
12.	To test the Large Cube. Select: Cube Select: Large Then Press Enter Then Press Clear after done until blank.		PASS
13.	To test the Small Triangle. Select: Triangle Select: Small Then Press Enter Then Press Clear after done until blank.		PASS

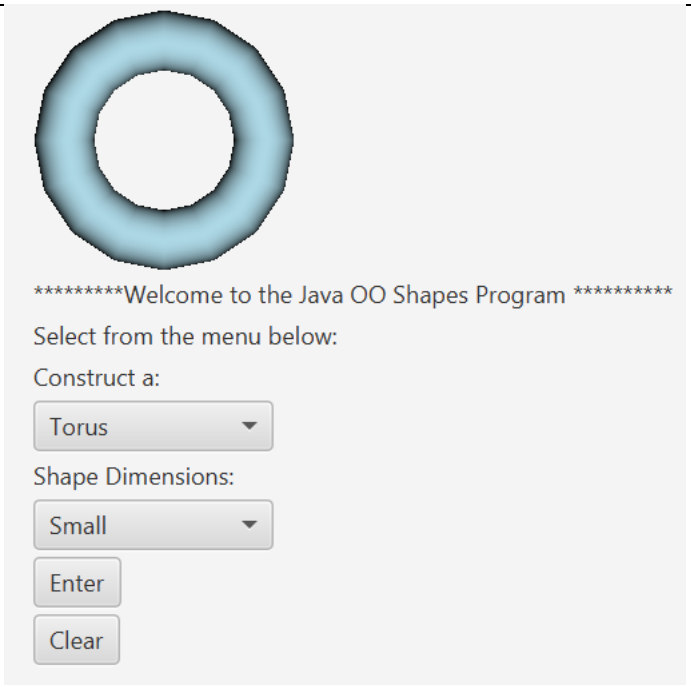
14.	To test the Medium Triangle. Select: Triangle Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS
15.	To test the Large Triangle. Select: Triangle Select: Large Then Press Enter Then Press Clear after done until blank.		PASS

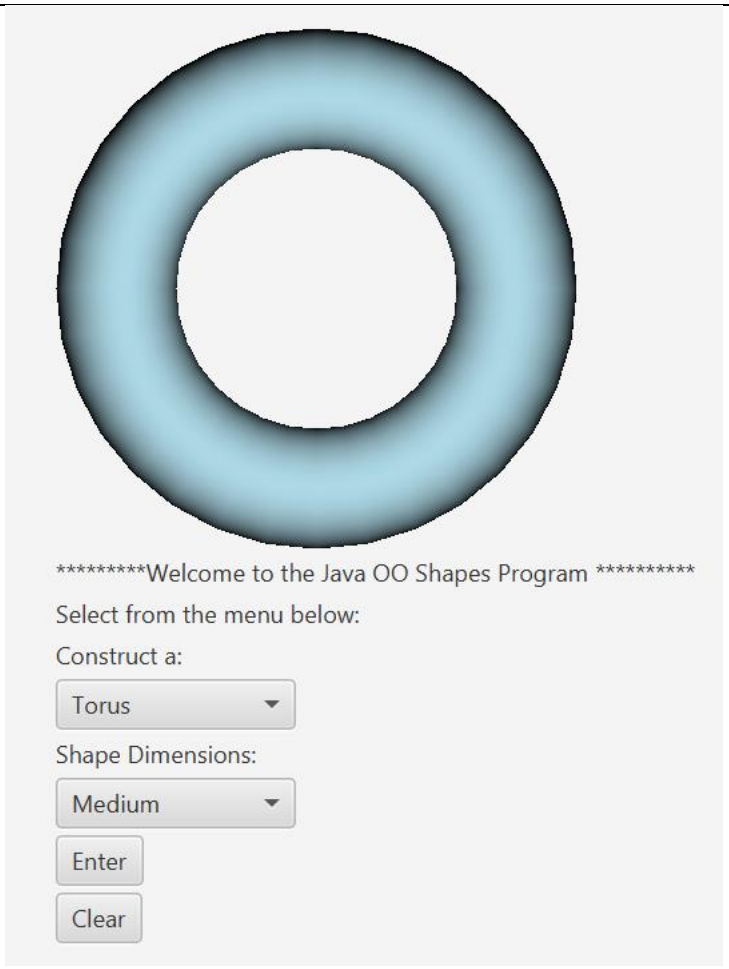
16.	To test the Small Sphere. Select: Sphere Select: Small Then Press Enter Then Press Clear after done until blank.		PASS
17.	To test the Medium Sphere. Select: Sphere Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS

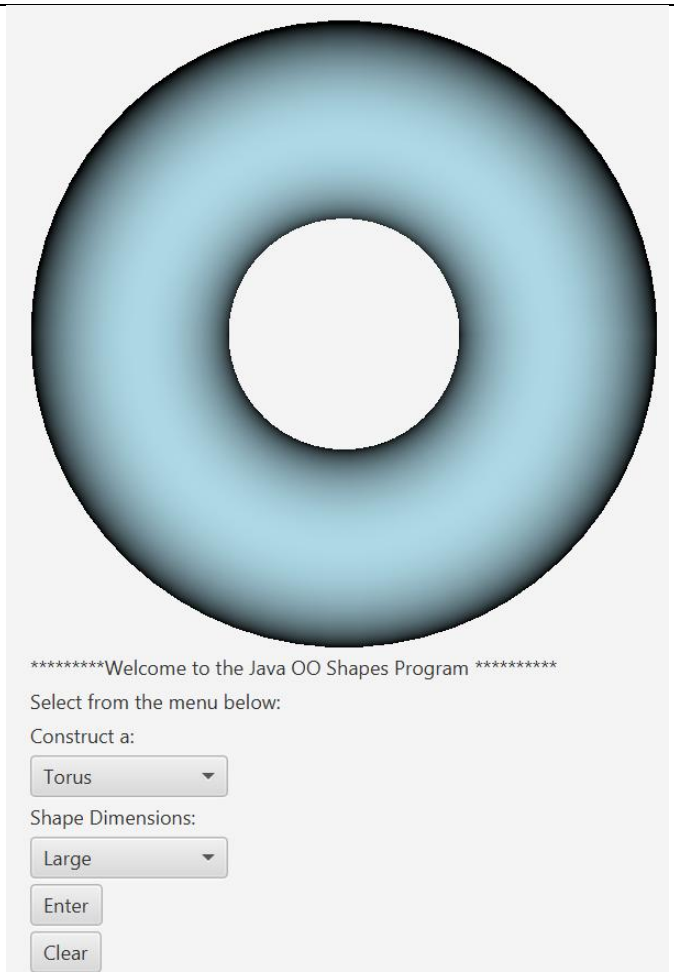
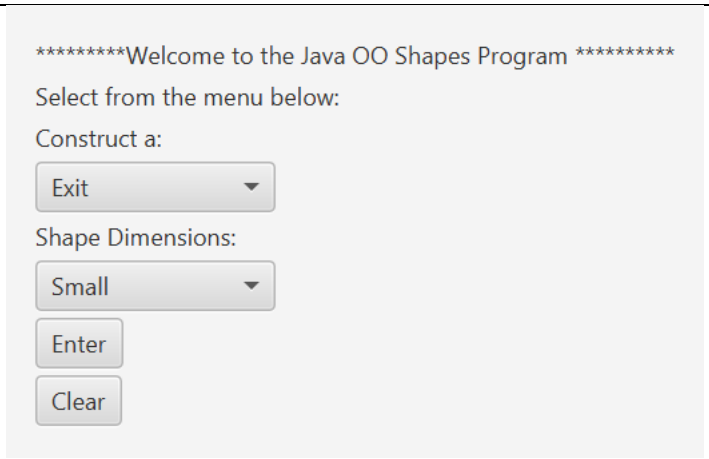
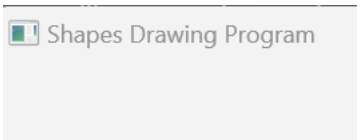
18.	To test the Large Sphere. Select: Sphere Select: Large Then Press Enter Then Press Clear after done until blank.		PASS
19.	To test the Small Cone. Select: Cone Select: Small Then Press Enter Then Press Clear after done until blank.		PASS

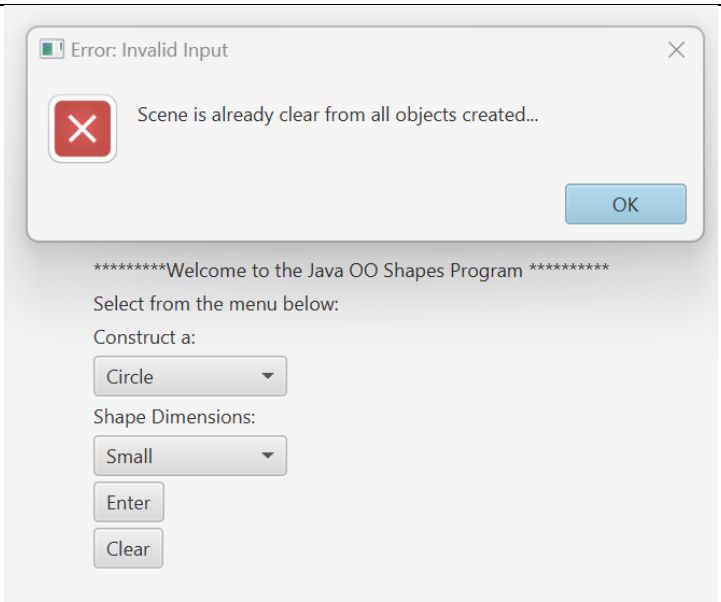
20.	To test the Medium Cone. Select: Cone Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS
21.	To test the Large Cone. Select: Cone Select: Large Then Press Enter Then Press Clear after done until blank.		PASS

22.	To test the Small Cylinder. Select: Cylinder Select: Small Then Press Enter Then Press Clear after done until blank.		PASS
23.	To test the Medium Cylinder. Select: Cylinder Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS

24.	To test the Large Cylinder. Select: Cylinder Select: Large Then Press Enter Then Press Clear after done until blank.		PASS
25.	To test the Small Torus. Select Torus Select: Small Then Press Enter Then Press Clear after done until blank.		PASS

26.	To test the Medium Torus. Select Torus Select: Medium Then Press Enter Then Press Clear after done until blank.		PASS

27.	To test the Large Torus. Select Torus Select: Large Then Press Enter Then Press Clear after done until blank.		PASS
28.	To test the Exit. Select Exit Then Press Enter It should close the program.	 	PASS

29.	To test the Clear button if the user tries to clear again, even though the area is already cleared (blank canvas and no shapes.) Press Clear when the canvas has no shapes, drawings, or images. It should pop up an alert that the user cannot clear it anymore as it is already blank. This prevents the user from clearing a blank canvas and exceeding the index bounds.		PASS

Lessons Learned (brief paragraphs):

To achieve my project goals, I learned how to create a project, classes, immutable classes, generics, try/catch, throw exceptions, JavaFX, encapsulation, inheritance, information hiding, polymorphism, methods, and functions for object-oriented programming for the Shapes Drawing Program. I learned that the Shapes Drawing Program allowed me to comprehend the significant principles of object-oriented programming and software development. A class defines all the attributes an object can have and methods that define the object's functionality. A subclass inherits the properties and behaviors of another class. Immutable classes in Java mean that once an object is created, we cannot change its content. And so, the main lesson learned from the Project1 class is that it is responsible for creating the Shapes Program Selection and the menu. This menu option and the follow-up combo boxes are used to find the shape type and size. I had to utilize the Combobox and buttons to produce the drawing or image. The Labeled class is the base class for Label, Button, etc. The ButtonBase class defines the onAction property for specifying a handler for action events. The TextInputControl class is for TextField which fires an action event if you code it. The classes have drawings embedded with the color to color the shape. The extra methods are the area for 2D shapes and the area and volume for 3D shapes. Moreover, these immutable classes are MyCircle, MySquare, MyTriangle, MyRectangle, MySphere, MyCube, MyCone, MyCone2, MyCylinder, and MyTorus. The main lesson learned from the Project2 class is that it is responsible for creating the user interface GUI. The main goals in this class's design include its GUI constructor, which initializes the object, and its methods so that the objects and sizes to be drawn or projected through an image by the user input. These lessons helped me understand good modular design for developing object-oriented programming with shapes and geometry. In real life, users can utilize this application to create shape drawings in Java GUI or even object animations. Project 2 is about comprehending Java GUI for JavaFX, the classes, hierarchy,

inheritance, encapsulation, information hiding, polymorphism, is-a, has-a, relationships, and subclasses. Overall, I learned to apply it to Project 2 with the lessons about JavaFX, try/catch, classes, subclasses, packages, importing libraries, constructors, object-oriented programming, encapsulation, inheritance, information hiding, and polymorphism.

My design approach was to create all the required classes before implementing the Project1 class. I started with a Bottom-Up Design when building the code, but then debugged the code through a Top-Down Design. I followed the instructions on what is asked for Project 2 and the other shapes classes. I utilized the lessons to apply them to Project 2. Once it was finished, I went back into the Project 2 class to create the Shapes program according to the rubric, and then the user inputs would be passed and the results back through the Project 2 class. Project 2 class creates a Shape selection menu program. This allows the user to input the shape type and size to project the drawing or image of the shape. To debug Project 2, I looked at the lessons, my old codes, and online concepts about this chapter. I then modified the classes. Then, I checked back to see if the output was correct through the Project 2 class.

References

Oracle. (n.d.). *The Java™ tutorials: Object-oriented programming concepts*. Oracle. https://docs.oracle.com/javase/tutorial/java/javaOO/index.html

Oracle. (n.d.). *Java Lesson: Subclassing*. Oracle. https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html

Oracle. (n.d.). *JavaFX API Documentation*. Oracle. https://docs.oracle.com/javafx/2/api/