CMSC 335

Project 2

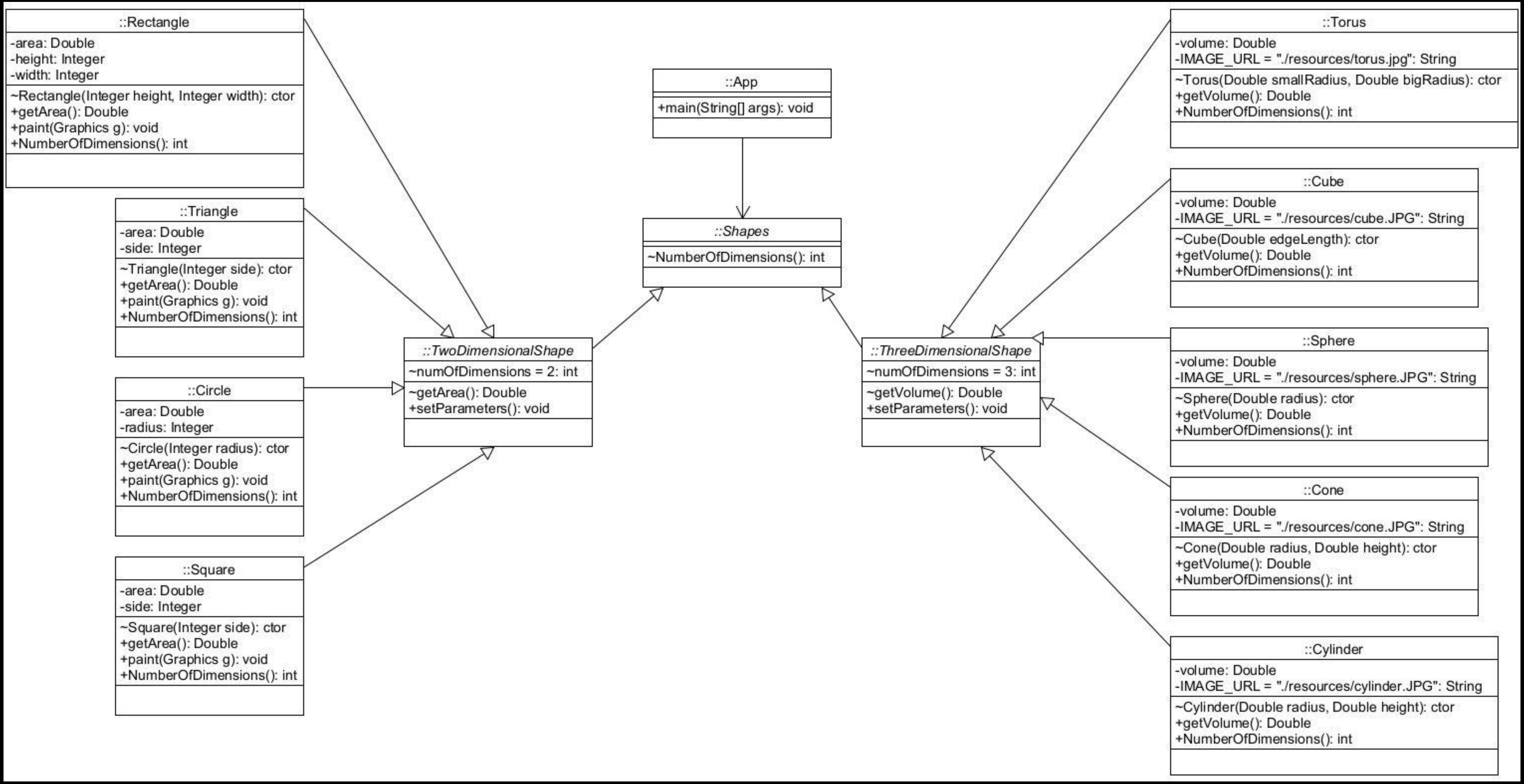
Allen Taylor

3/27/2022

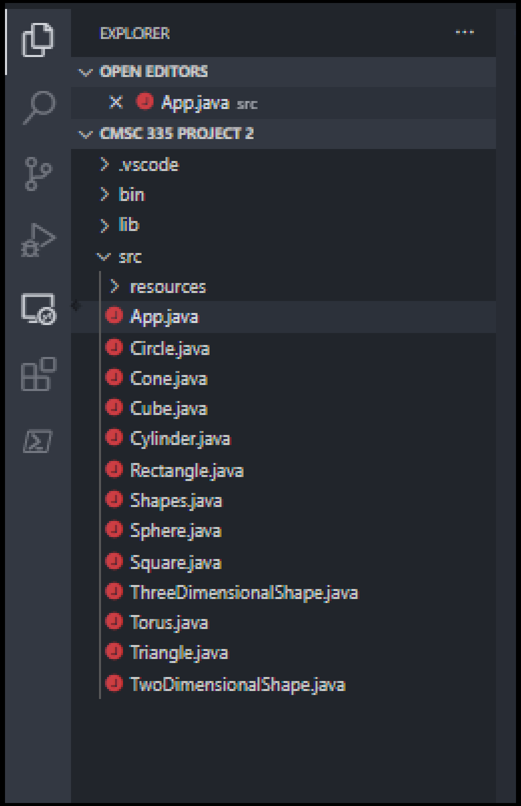
Requirements:

- Design, implement and test a set of Java classes that allows a user to select a shape from a list of available shapes, enter appropriate dimensional parameters and then display that shape in a frame of your Swing-based GUI. For 3-D shapes consider loading an image from a file and displaying that as a representative.
- Designs a Java class Inheritance hierarchy that would satisfy the following is-a and has-a relationships:
  - Circle
  - Square
  - Triangle
  - Rectangle
  - Sphere
  - Cube
  - Cone
  - Cylinder
  - Torus

UML Diagram:

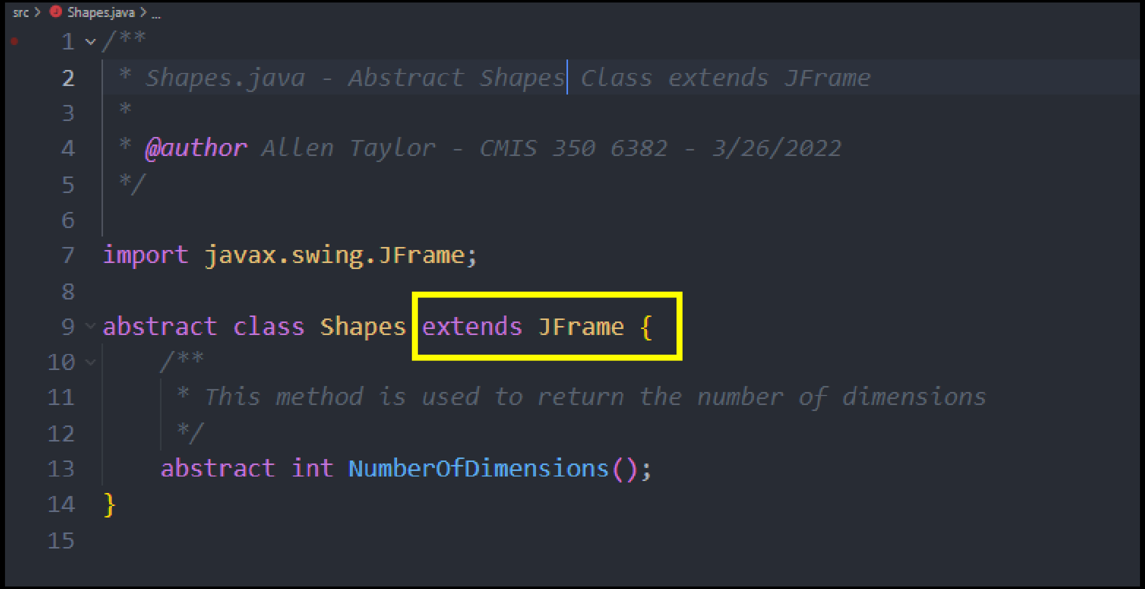

::Rectangle
-area: Double
-height: Integer
-width: Integer
~Rectangle(Integer height, Integer width): ctor
+getArea(): Double
+paint(Graphics g): void
+NumberOfDimensions(): int
::Triangle
-area: Double
-side: Integer
~Triangle(Integer side): ctor
+getArea(): Double
+paint(Graphics g): void
+NumberOfDimensions(): int
::Circle
-area: Double
-radius: Integer
~Circle(Integer radius): ctor
+getArea(): Double
+paint(Graphics g): void
+NumberOfDimensions(): int
::Square
-area: Double
-side: Integer
~Square(Integer side): ctor
+getArea(): Double
+paint(Graphics g): void
+NumberOfDimensions(): int
::App
+main(String[] args): void
::Shapes
~NumberOfDimensions(): int
::TwoDimensionalShape
~numOfDimensions = 2: int
~getArea(): Double
+setParameters(): void
::ThreeDimensionalShape
~numOfDimensions = 3: int
~getVolume(): Double
+setParameters(): void
::Torus
-volume: Double
-IMAGE_URL = "./resources/torus.jpg": String
~Torus(Double smallRadius, Double bigRadius): ctor
+getVolume(): Double
+NumberOfDimensions(): int
::Cube
-volume: Double
-IMAGE_URL = "./resources/cube.JPG": String
~Cube(Double edgeLength): ctor
+getVolume(): Double
+NumberOfDimensions(): int
::Sphere
-volume: Double
-IMAGE_URL = "./resources/sphere.JPG": String
~Sphere(Double radius): ctor
+getVolume(): Double
+NumberOfDimensions(): int
::Cone
-volume: Double
-IMAGE_URL = "./resources/cone.JPG": String
~Cone(Double radius, Double height): ctor
+getVolume(): Double
+NumberOfDimensions(): int
::Cylinder
-volume: Double
-IMAGE_URL = "./resources/cylinder.JPG": String
~Cylinder(Double radius, Double height): ctor
+getVolume(): Double
+NumberOfDimensions(): int

File Directory:



```
EXPLORER                                ···
∨ OPEN EDITORS
   × App.java src
∨ CMSC 335 PROJECT 2
  > .vscode
  > bin
  > lib
  ∨ src
    > resources
      App.java
      Circle.java
      Cone.java
      Cube.java
      Cylinder.java
      Rectangle.java
      Shapes.java
      Sphere.java
      Square.java
      ThreeDimensionalShape.java
      Torus.java
      Triangle.java
      TwoDimensionalShape.java
```

Shapes Class:


src > Shapes.java > ...

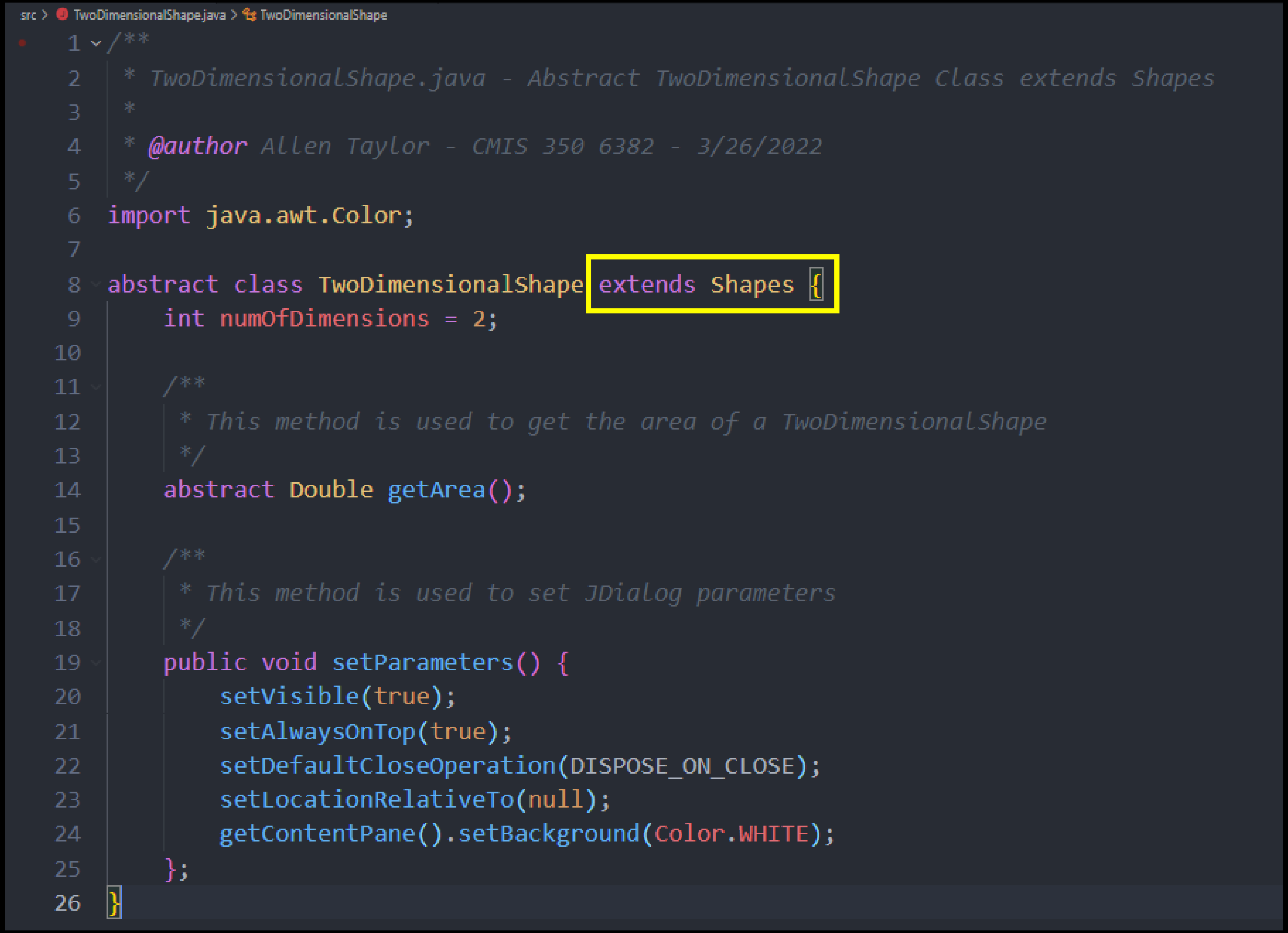
```java
/**
 * Shapes.java - Abstract Shapes Class extends JFrame
 *
 * @author Allen Taylor - CMIS 350 6382 - 3/26/2022
 */

import javax.swing.JFrame;

abstract class Shapes extends JFrame {
    /**
     * This method is used to return the number of dimensions
     */
    abstract int NumberOfDimensions();
}

```

TwoDimensionalShape Class:



src > TwoDimensionalShape.java > TwoDimensionalShape

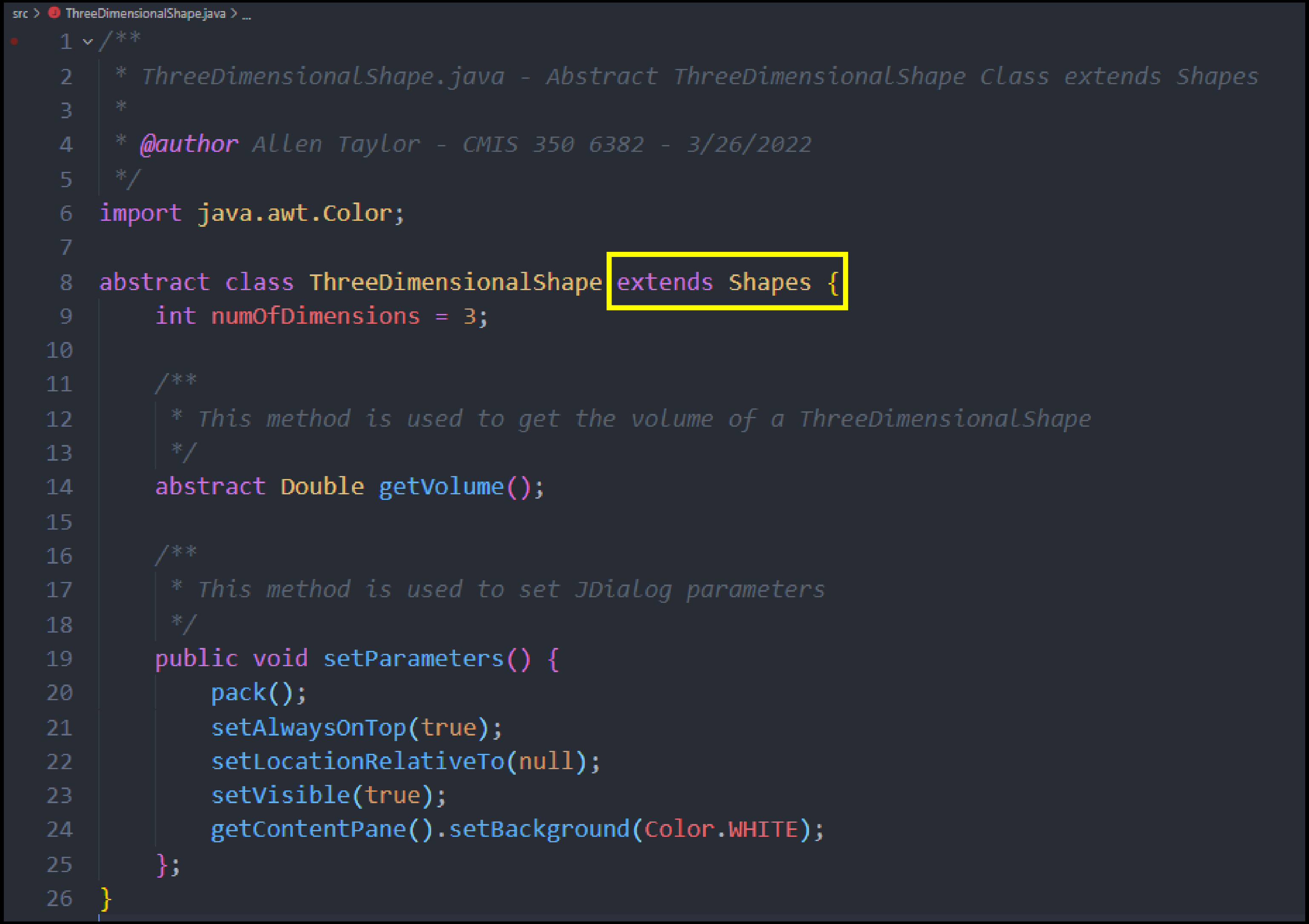
```java
/**
 * TwoDimensionalShape.java - Abstract TwoDimensionalShape Class extends Shapes
 *
 * @author Allen Taylor - CMIS 350 6382 - 3/26/2022
 */
import java.awt.Color;

abstract class TwoDimensionalShape extends Shapes {
    int numOfDimensions = 2;

    /**
     * This method is used to get the area of a TwoDimensionalShape
     */
    abstract Double getArea();

    /**
     * This method is used to set JDialog parameters
     */
    public void setParameters() {
        setVisible(true);
        setAlwaysOnTop(true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        getContentPane().setBackground(Color.WHITE);
    };
}
```

ThreeDimensionalShape Class:


src > ThreeDimensionalShape.java > ...

```java
/**
 * ThreeDimensionalShape.java - Abstract ThreeDimensionalShape Class extends Shapes
 *
 * @author Allen Taylor - CMIS 350 6382 - 3/26/2022
 */
import java.awt.Color;

abstract class ThreeDimensionalShape extends Shapes {
    int numOfDimensions = 3;

    /**
     * This method is used to get the volume of a ThreeDimensionalShape
     */
    abstract Double getVolume();

    /**
     * This method is used to set JDialog parameters
     */
    public void setParameters() {
        pack();
        setAlwaysOnTop(true);
        setLocationRelativeTo(null);
        setVisible(true);
        getContentPane().setBackground(Color.WHITE);
    };
}
```

Users Guide:

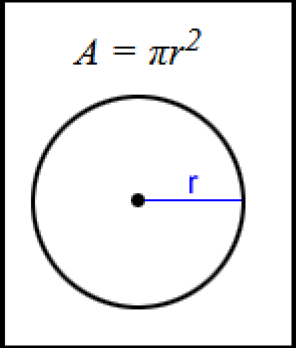
1. Select Shape from the **Select a Shape** combo box and click the **Create Shape** button.
2. Enter the required parameters for the shape.

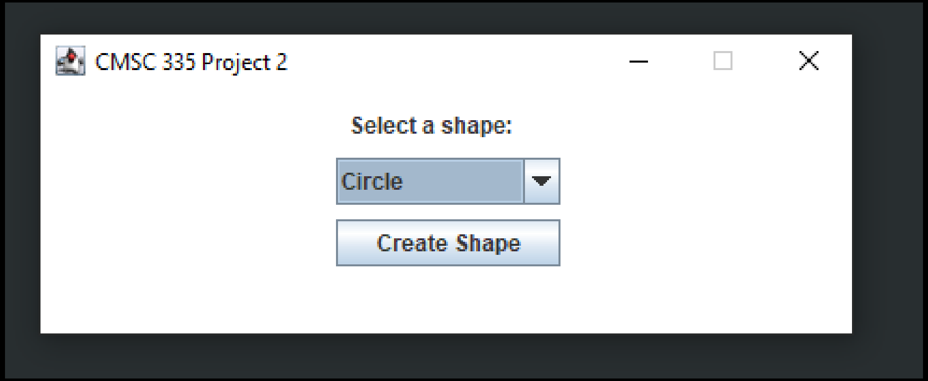
   For example: Radius, Length, Width, etc.
3. The shape will be displayed to the screen with the **Area** or **Volume** in the title bar.
4. Click exit to close the shape frame and repeat as desired.
5. Close the program when finished.

Construct a Circle:

Input:
400

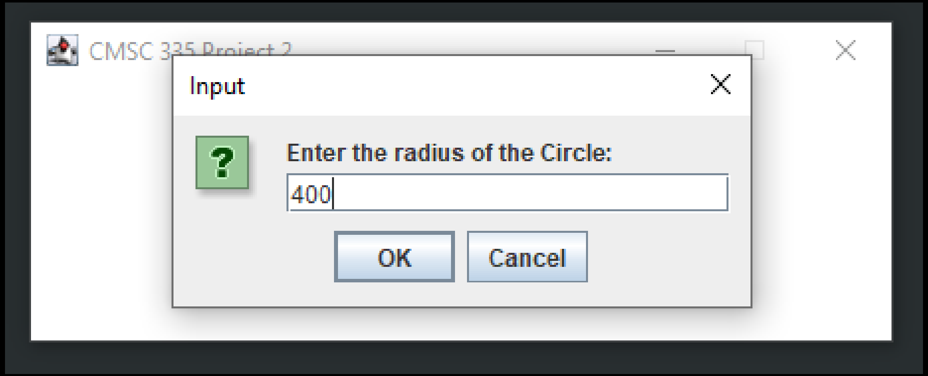
Expected Output:
Area = 502654.825




CMSC 335 Project 2

Select a shape:

Circle


CMSC 335 Project 2

Input

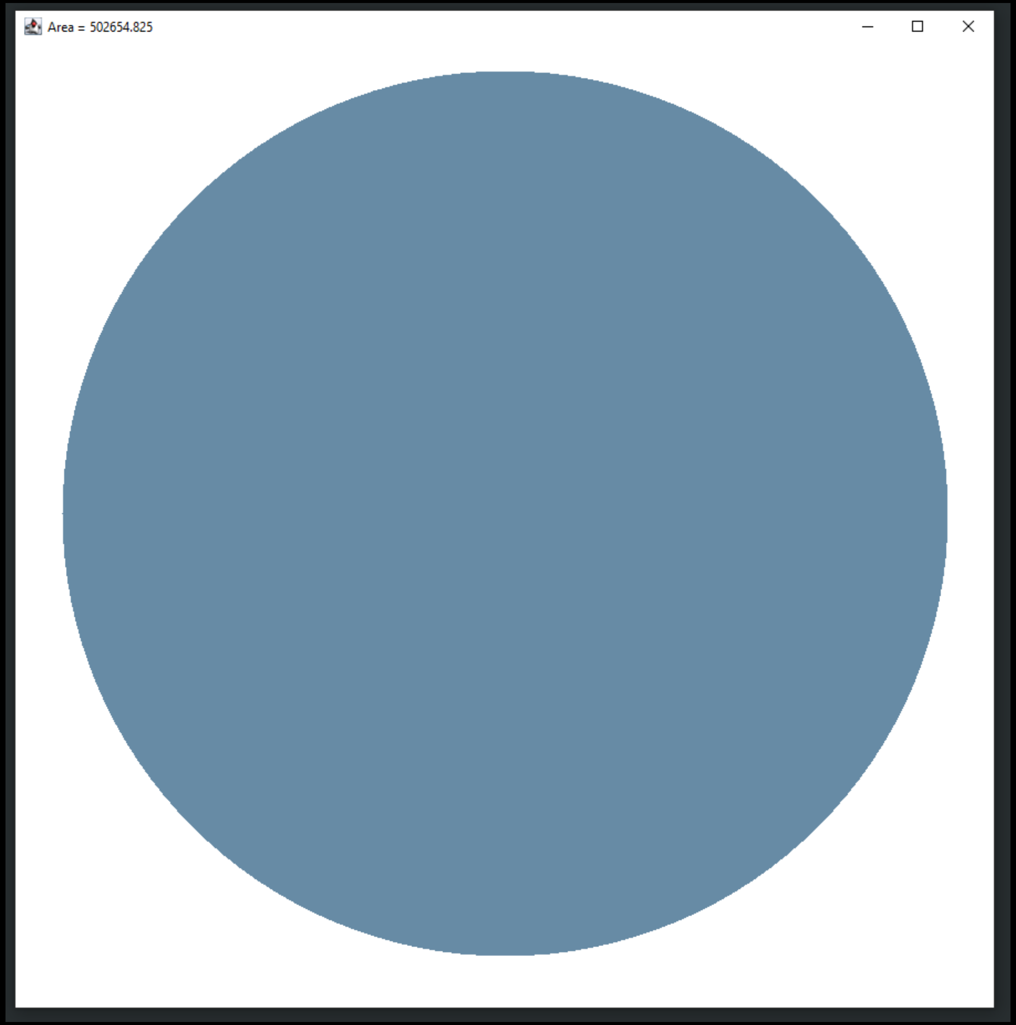
Enter the radius of the Circle:

400

Construct a Circle:

Actual Output:

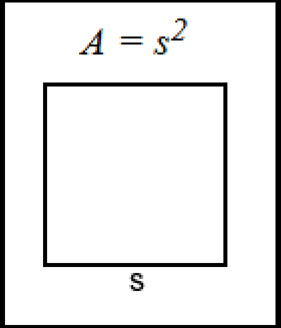
Area = 502654.825

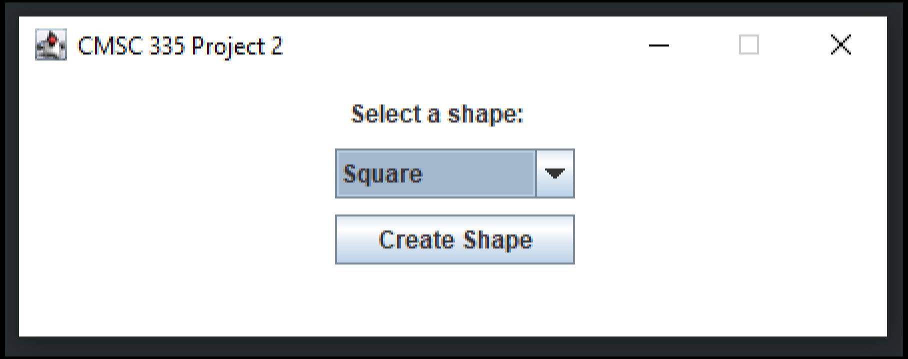

Area = 502654.825

Construct a Square:

Input:
400

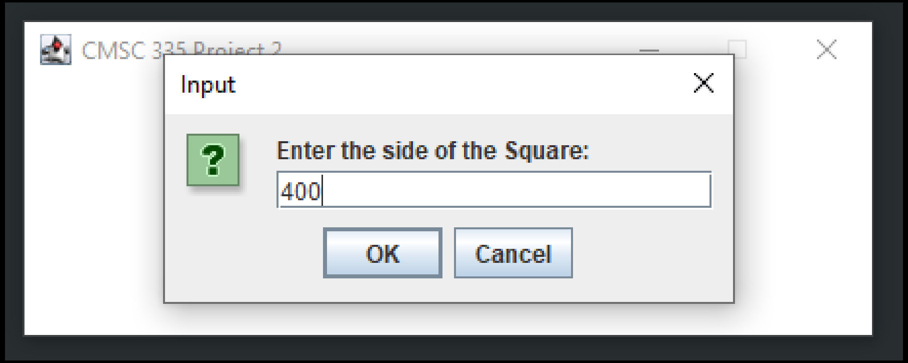
Expected Output:
Area = 160000




CMSC 335 Project 2

Select a shape:

Square


CMSC 335 Project 2

Input

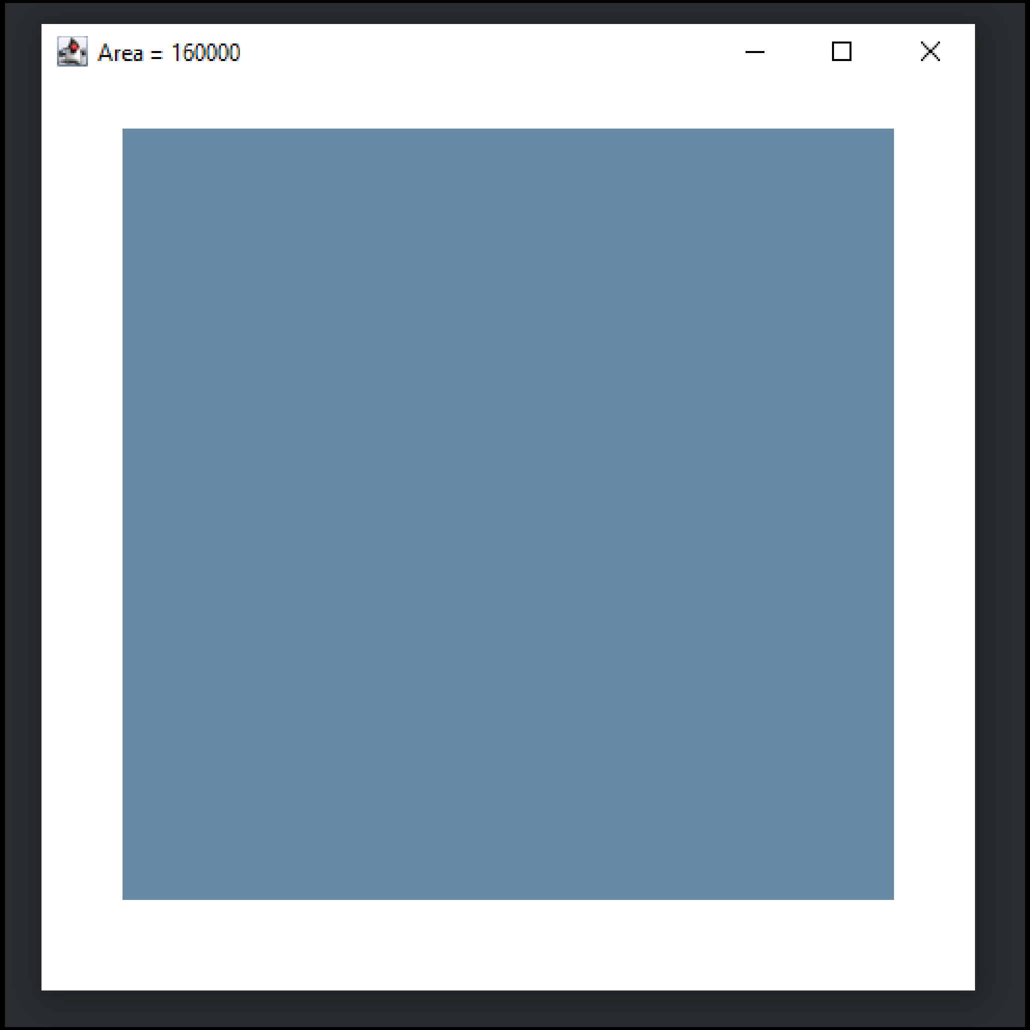
Enter the side of the Square:

400

Construct a Square:

Actual Output:

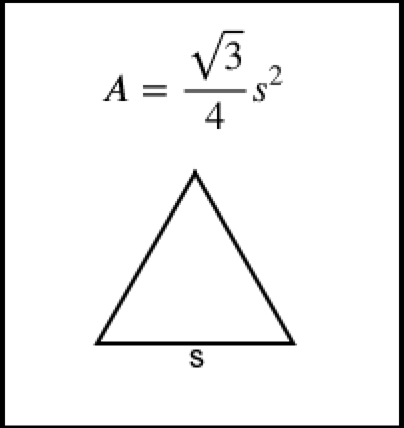
Area = 160000

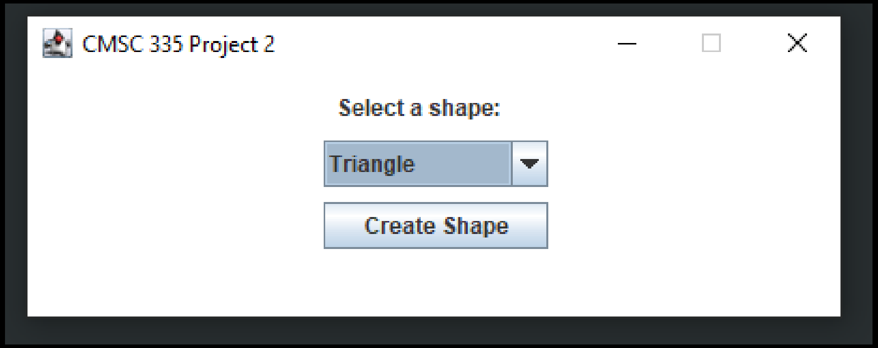


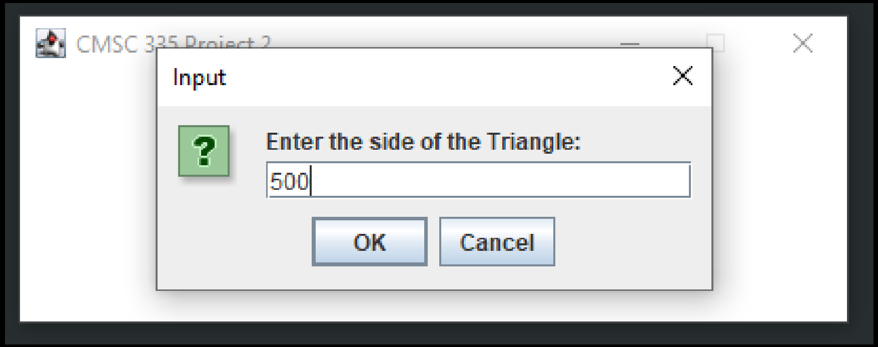
Construct a Triangle:

Input:
500

Expected Output:
Area = 108253.175

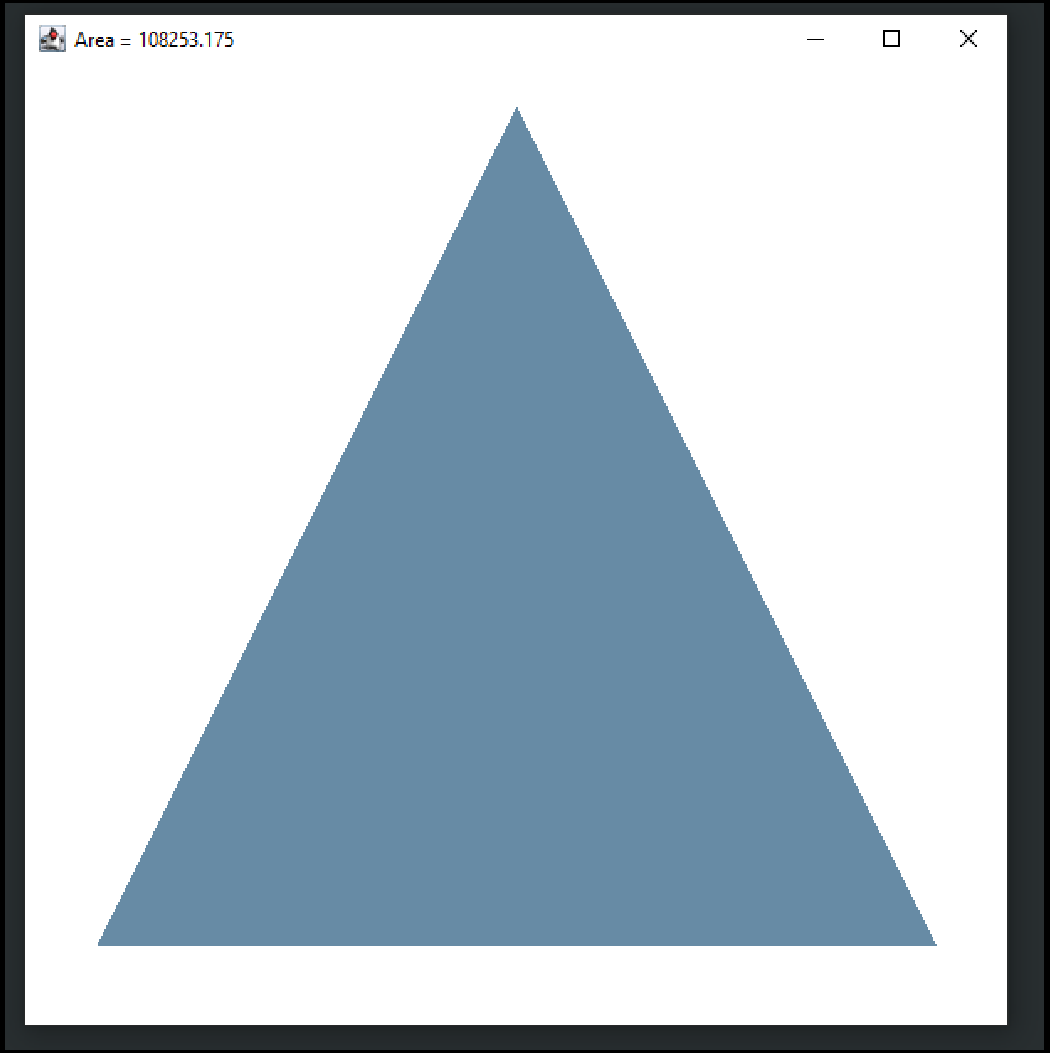

$$A = \frac{\sqrt{3}}{4}s^2$$





Construct a Triangle:

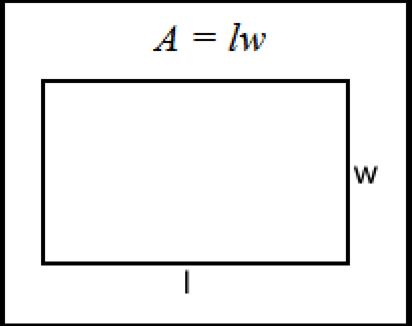
Actual Output:
Area = 108253.175

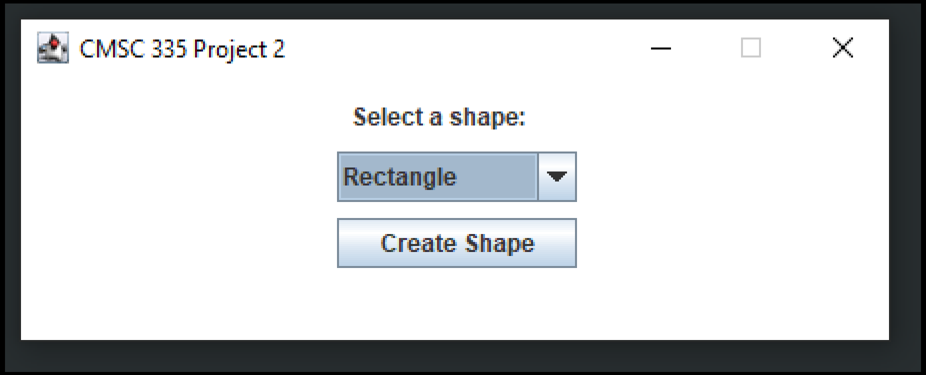


Construct a Rectangle:

Input:
400
600

Expected Output:
Area = 240000

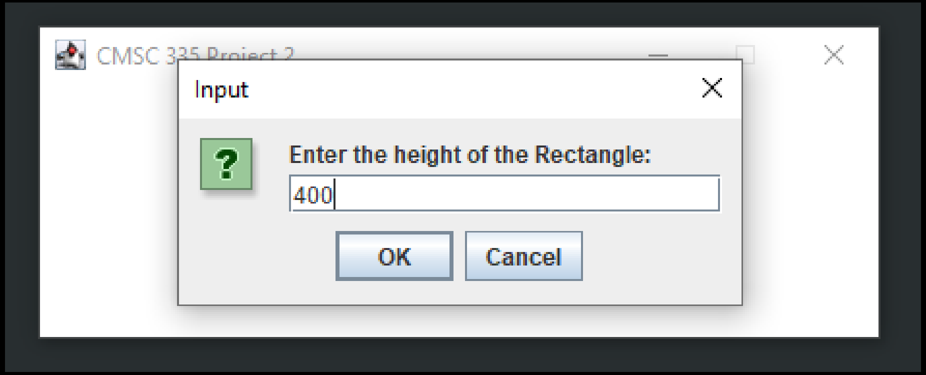



CMSC 335 Project 2

Select a shape:

Rectangle

Create Shape

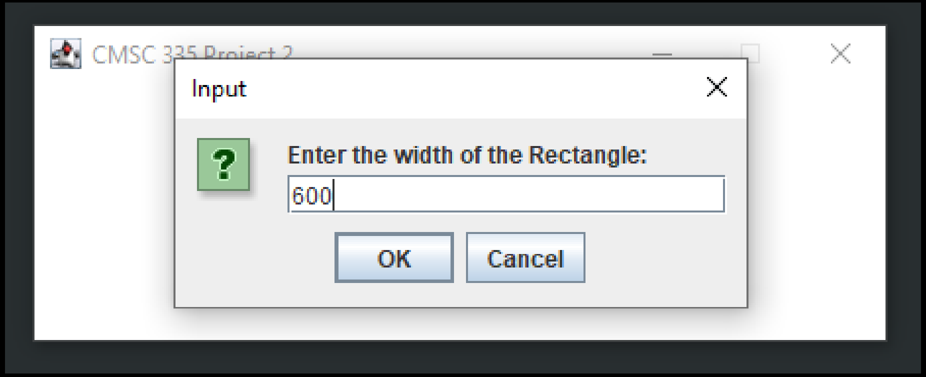

CMSC 335 Project 2

Input

Enter the height of the Rectangle:

400

OK Cancel


CMSC 335 Project 2

Input

Enter the width of the Rectangle:

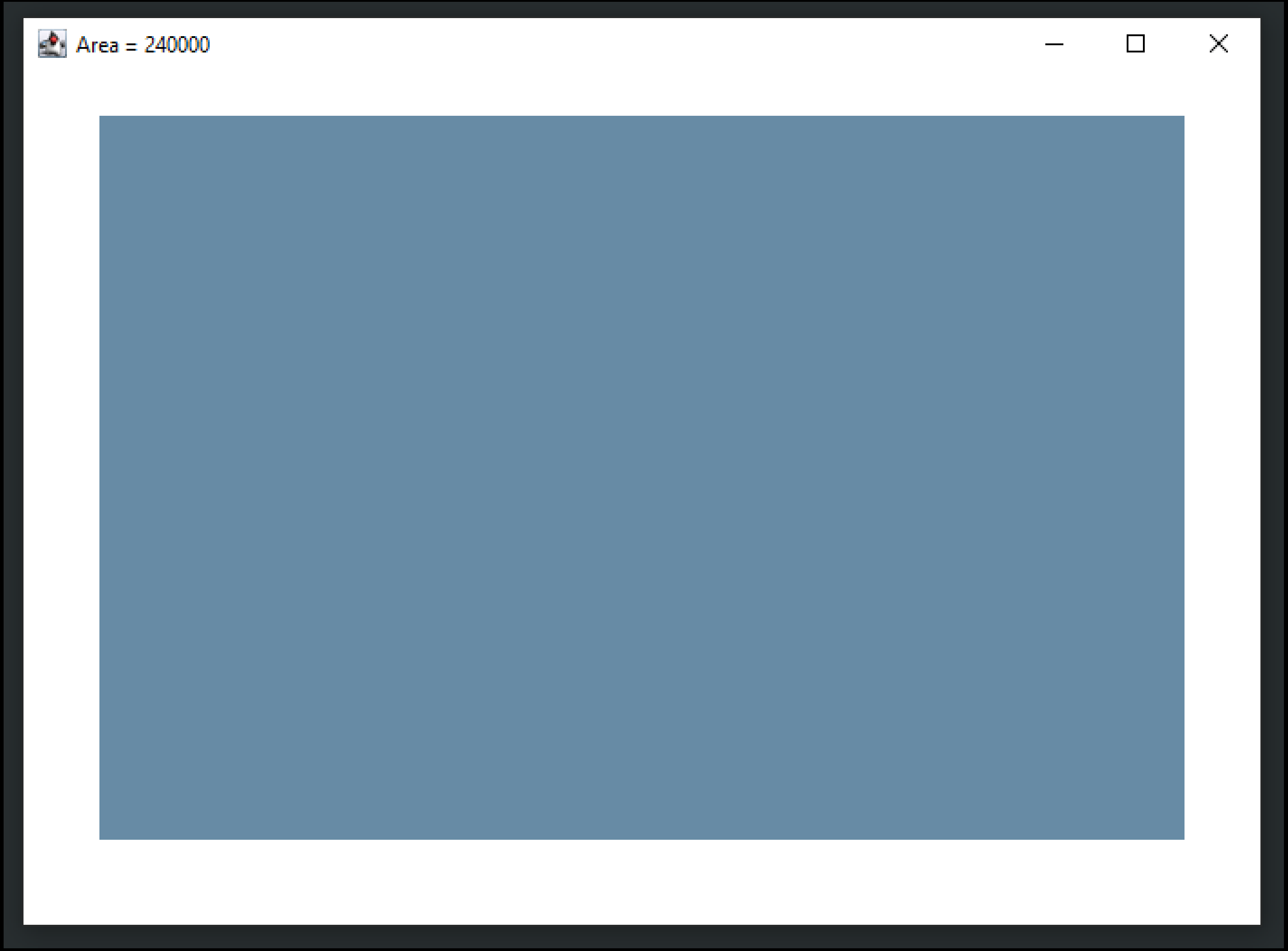
600

OK Cancel

Construct a Rectangle:

Actual Output:

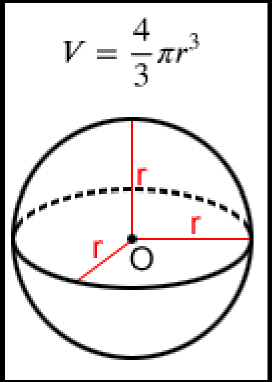
Area = 240000

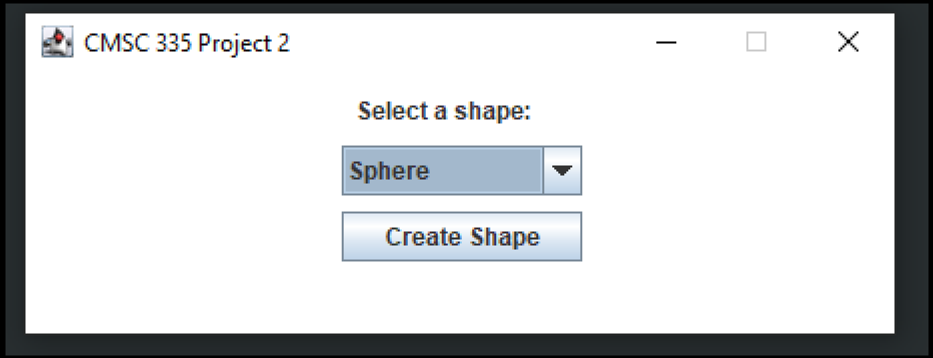


Construct a Sphere:

Input:
15

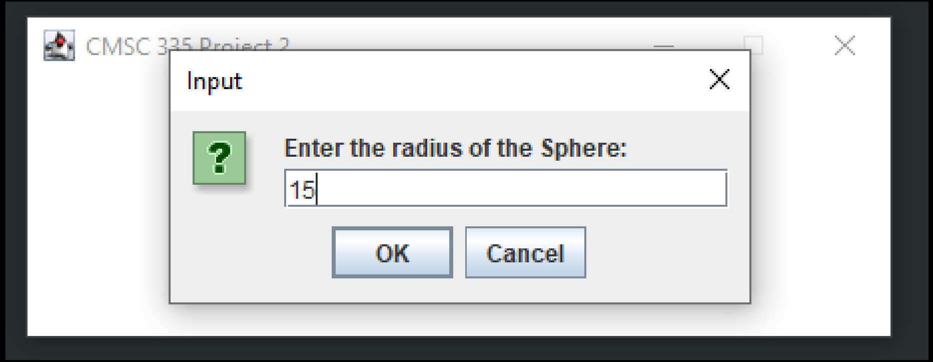
Expected Output:
Volume = 14137.167


$$V = \frac{4}{3}\pi r^3$$


CMSC 335 Project 2

Select a shape:

Sphere


CMSC 335 Project 2

Input

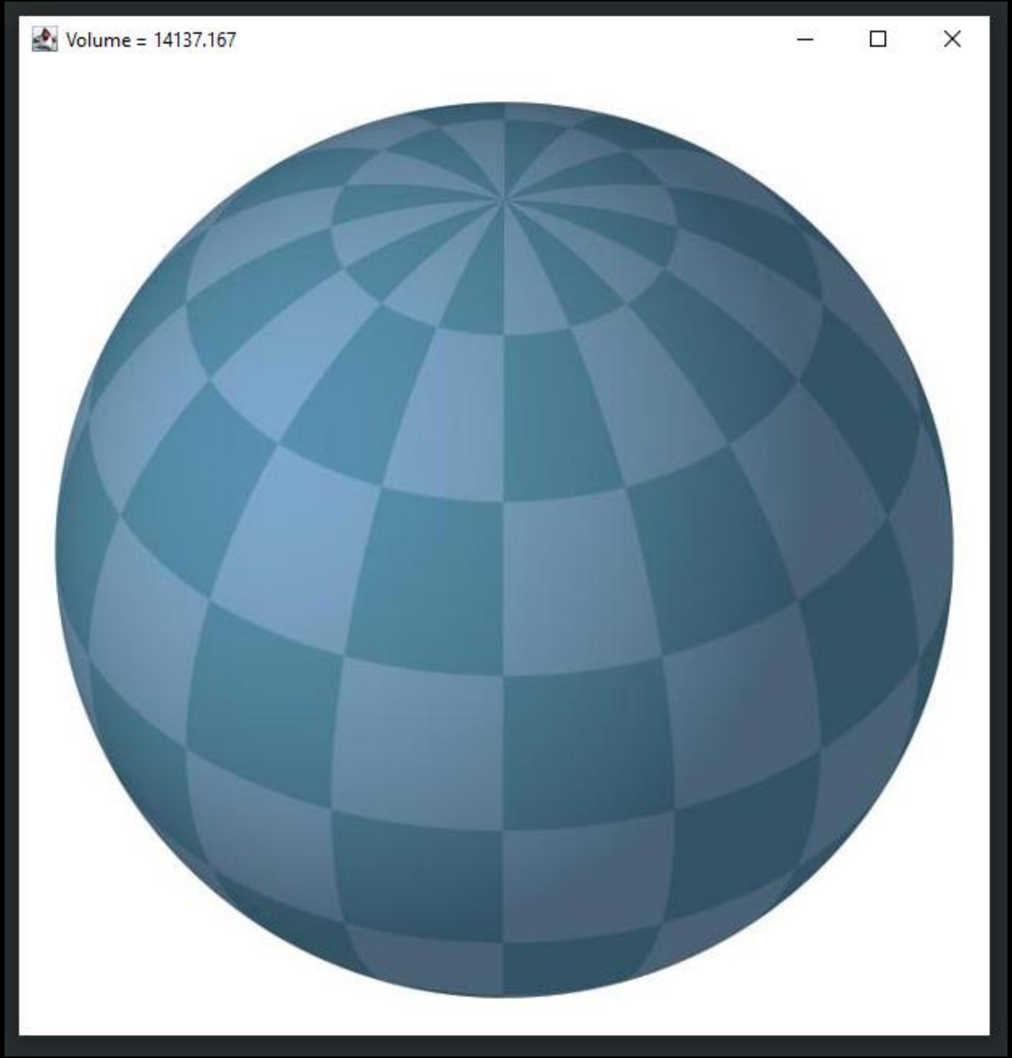
Enter the radius of the Sphere:

15

Construct a Sphere:

Actual Output:

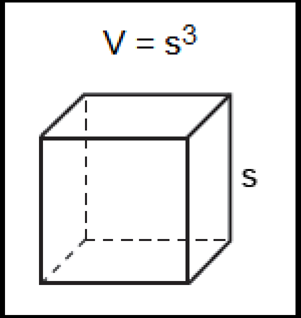
```
Volume = 14137.167
```

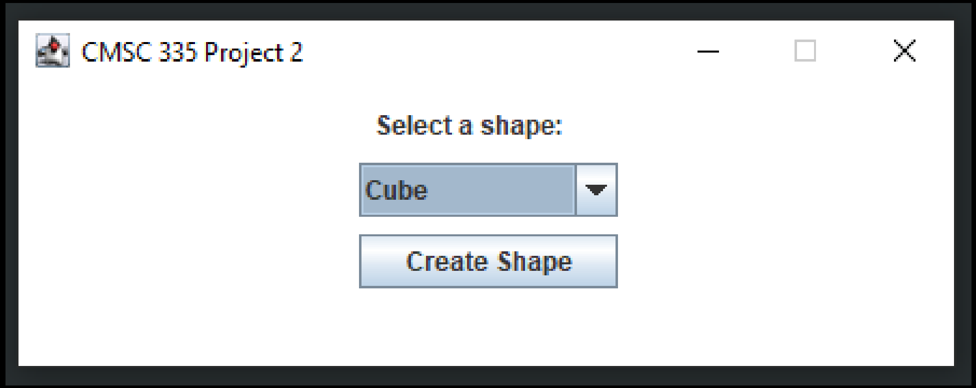


Construct a Cube:

Input:
5

Expected Output:
Volume = 125

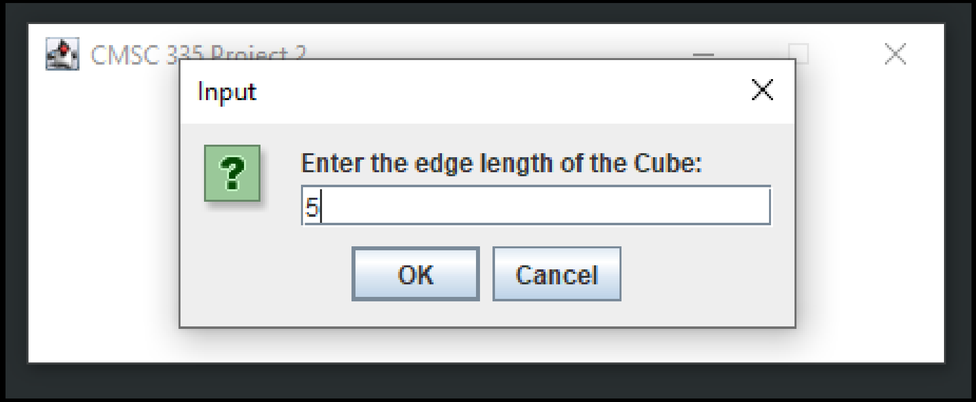



CMSC 335 Project 2

Select a shape:

Cube

Create Shape


CMSC 335 Project 2

Input

Enter the edge length of the Cube:

5

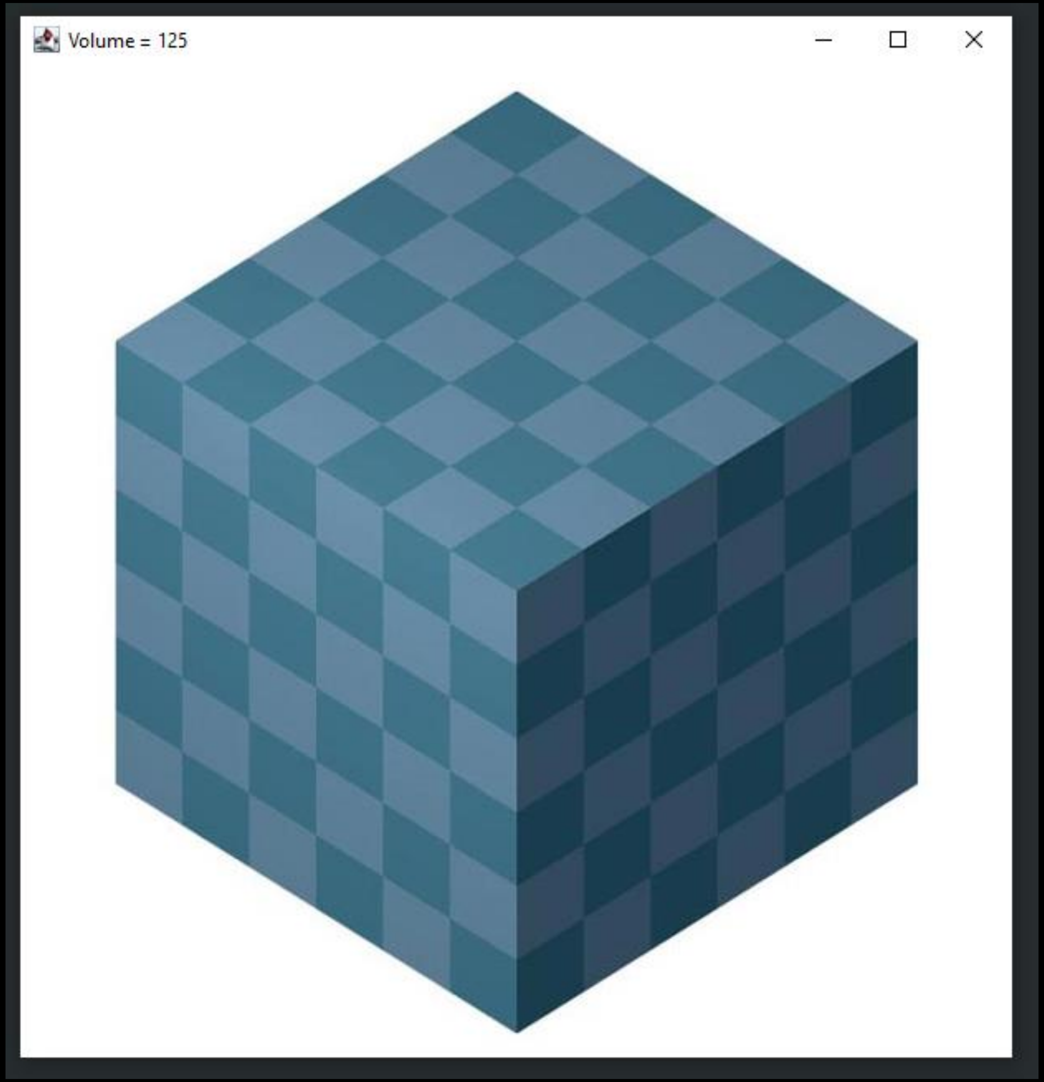
OK

Cancel

Construct a Cube:

Actual Output:

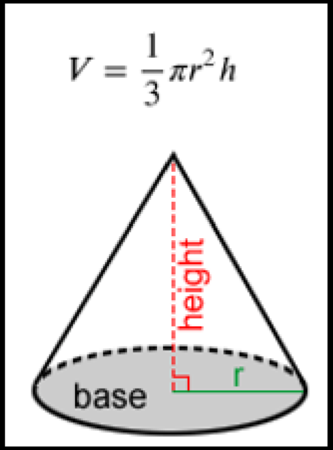
Volume = 125

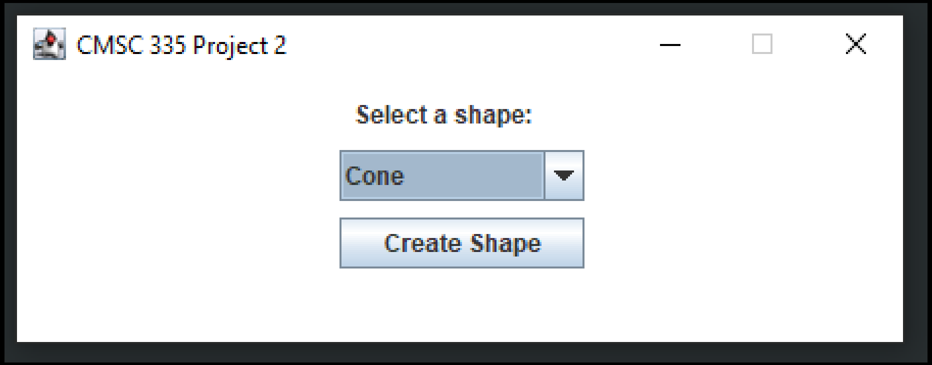


Construct a Cone:

Input:
5
10

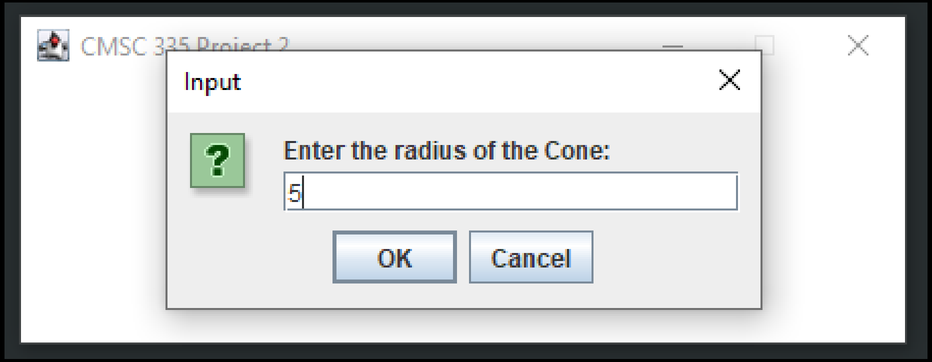
Expected Output:
Volume = 261.799


$$V = \frac{1}{3}\pi r^2 h$$


CMSC 335 Project 2

Select a shape:

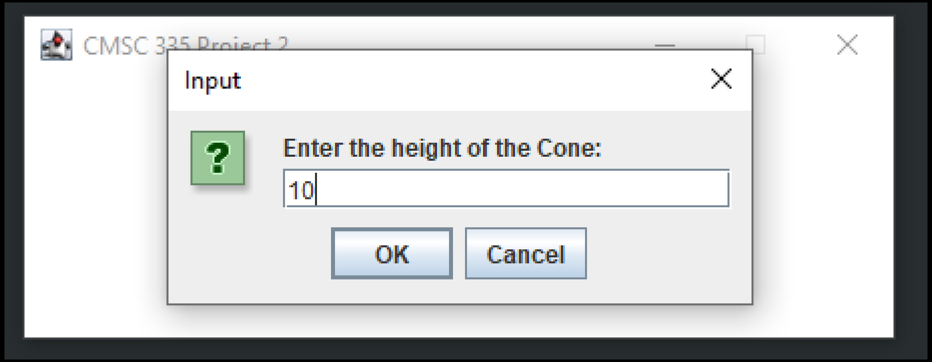
Cone


CMSC 335 Project 2

Input

Enter the radius of the Cone:

5


CMSC 335 Project 2

Input

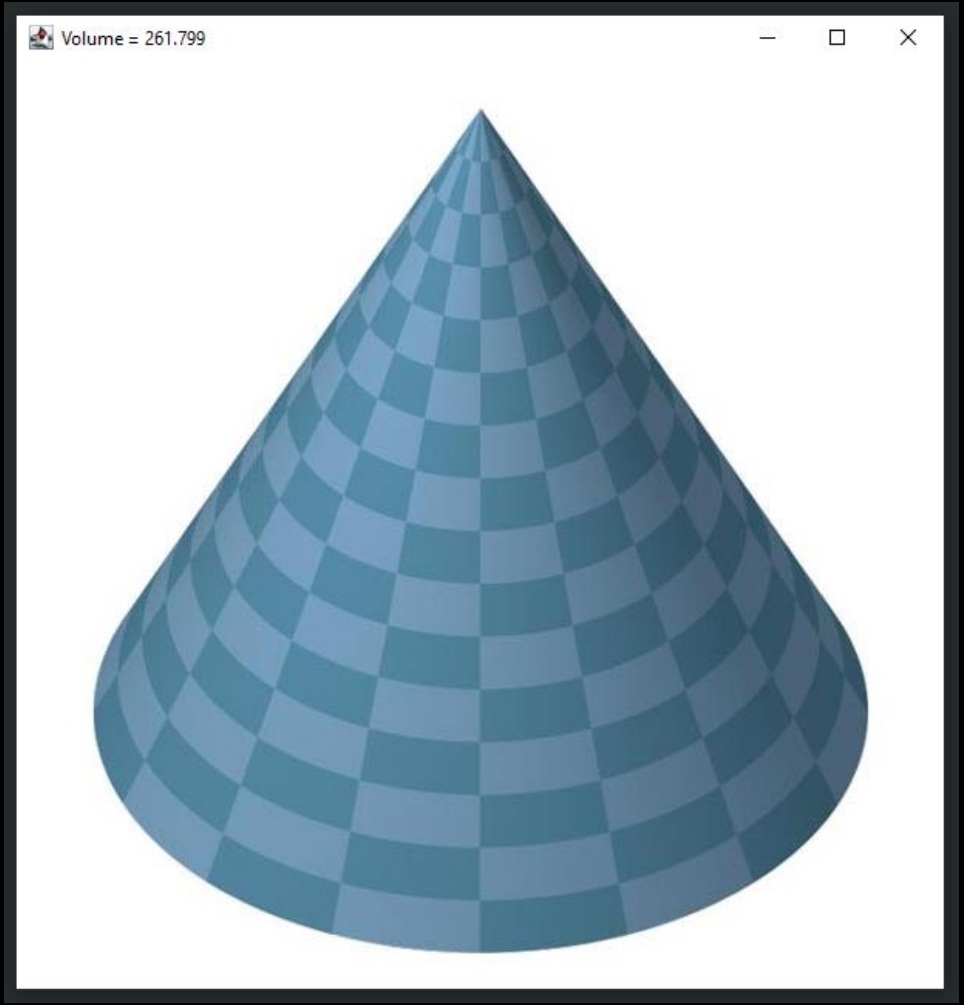
Enter the height of the Cone:

10

Construct a Cone:

Actual Output:

Volume = 261.799

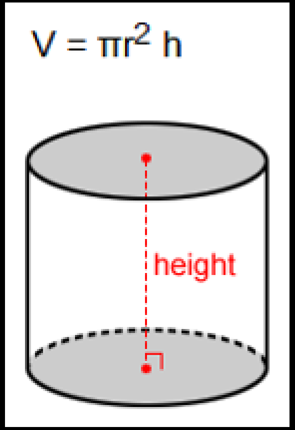


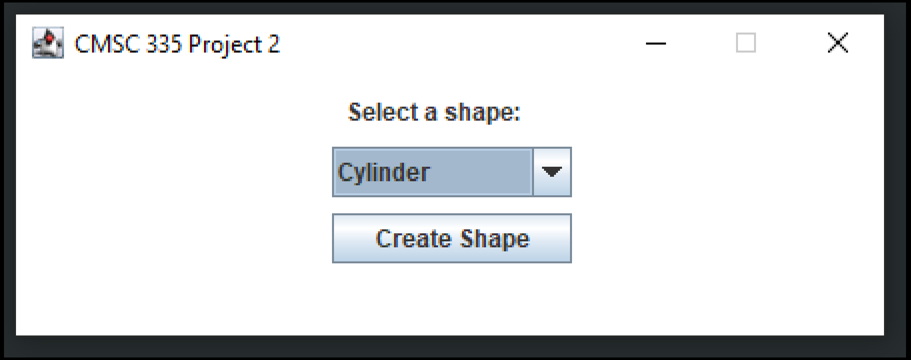
Volume = 261.799

Construct a Cylinder:

Input:
5
10

Expected Output:
Volume = 261.799

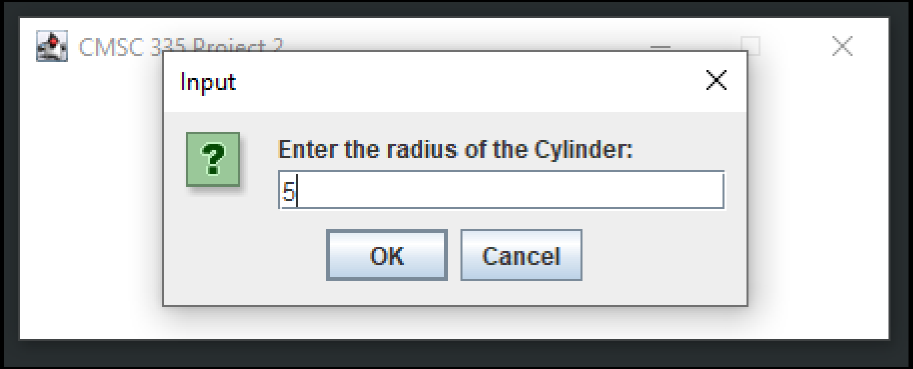



CMSC 335 Project 2

Select a shape:

Cylinder

Create Shape

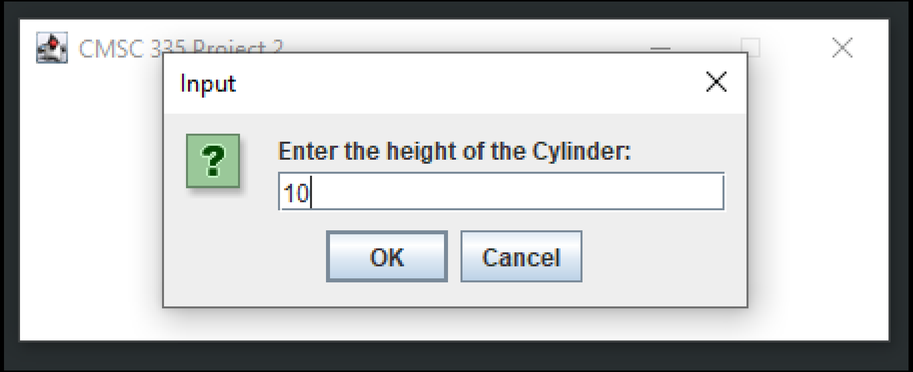

CMSC 335 Project 2

Input

Enter the radius of the Cylinder:

5

OK Cancel


CMSC 335 Project 2

Input

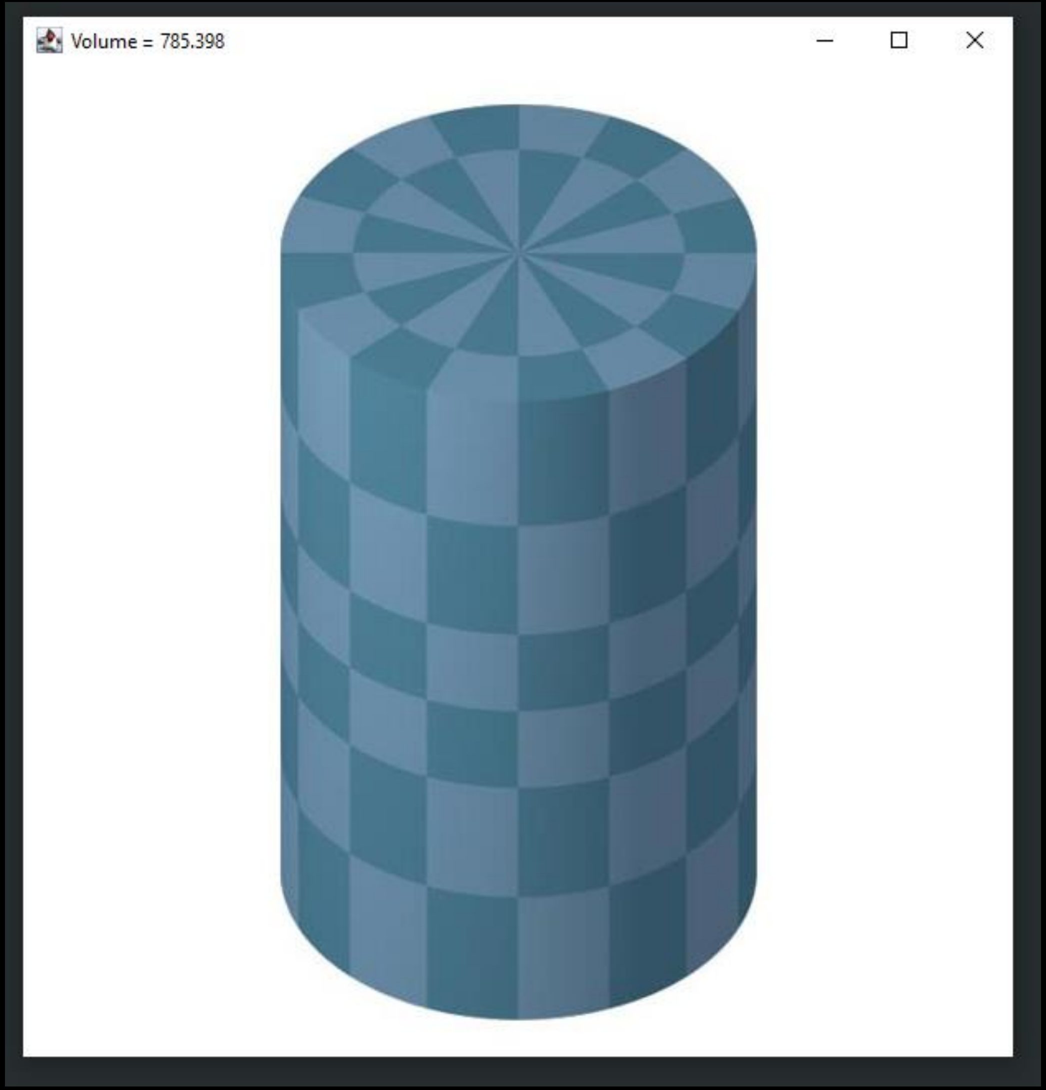
Enter the height of the Cylinder:

10

OK Cancel

Construct a Cylinder:

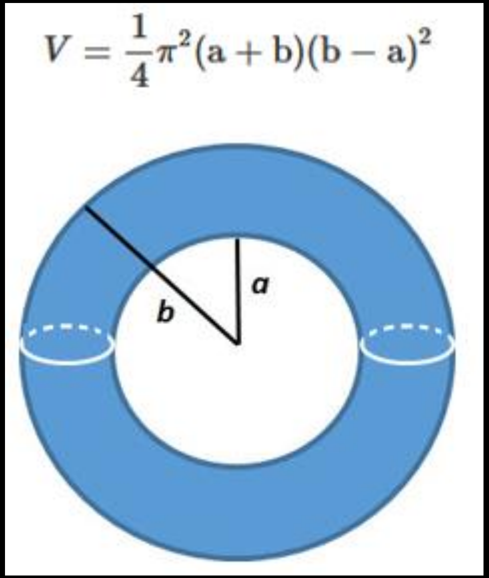
Actual Output:
Volume = 785.398

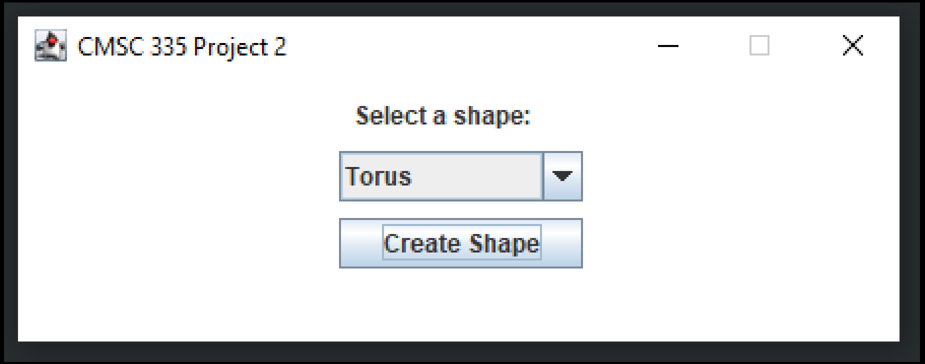


Construct a Torus:

Input:
3
6

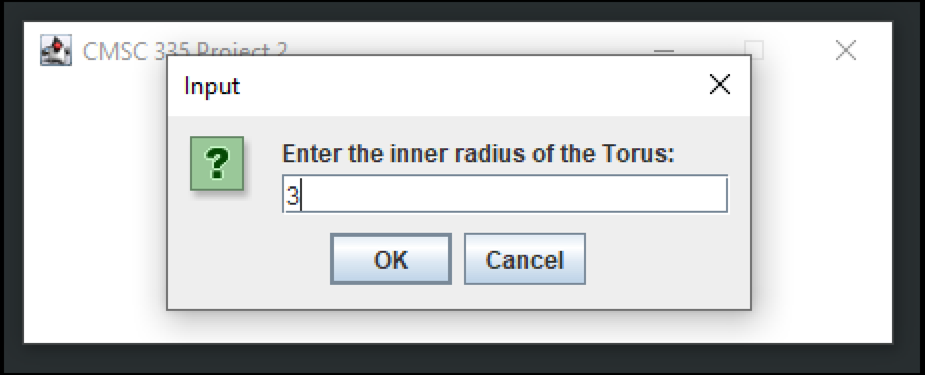
Expected Output:
Volume = 199.859


$$V = \frac{1}{4}\pi^2(a+b)(b-a)^2$$


CMSC 335 Project 2

Select a shape:

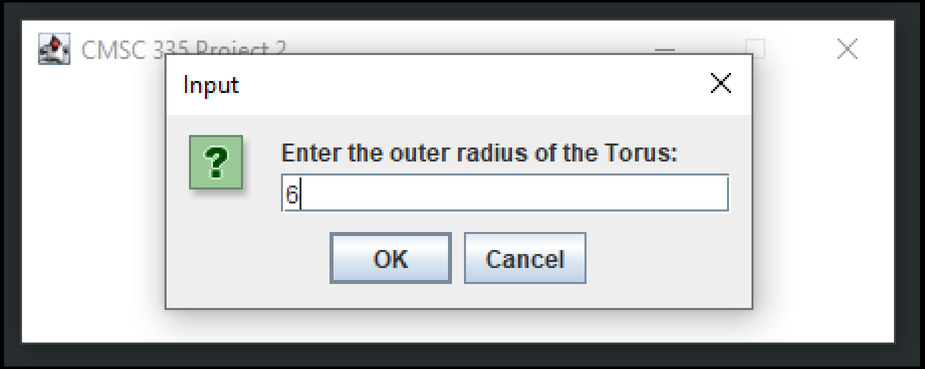
Torus


CMSC 335 Project 2

Input

Enter the inner radius of the Torus:

3


CMSC 335 Project 2

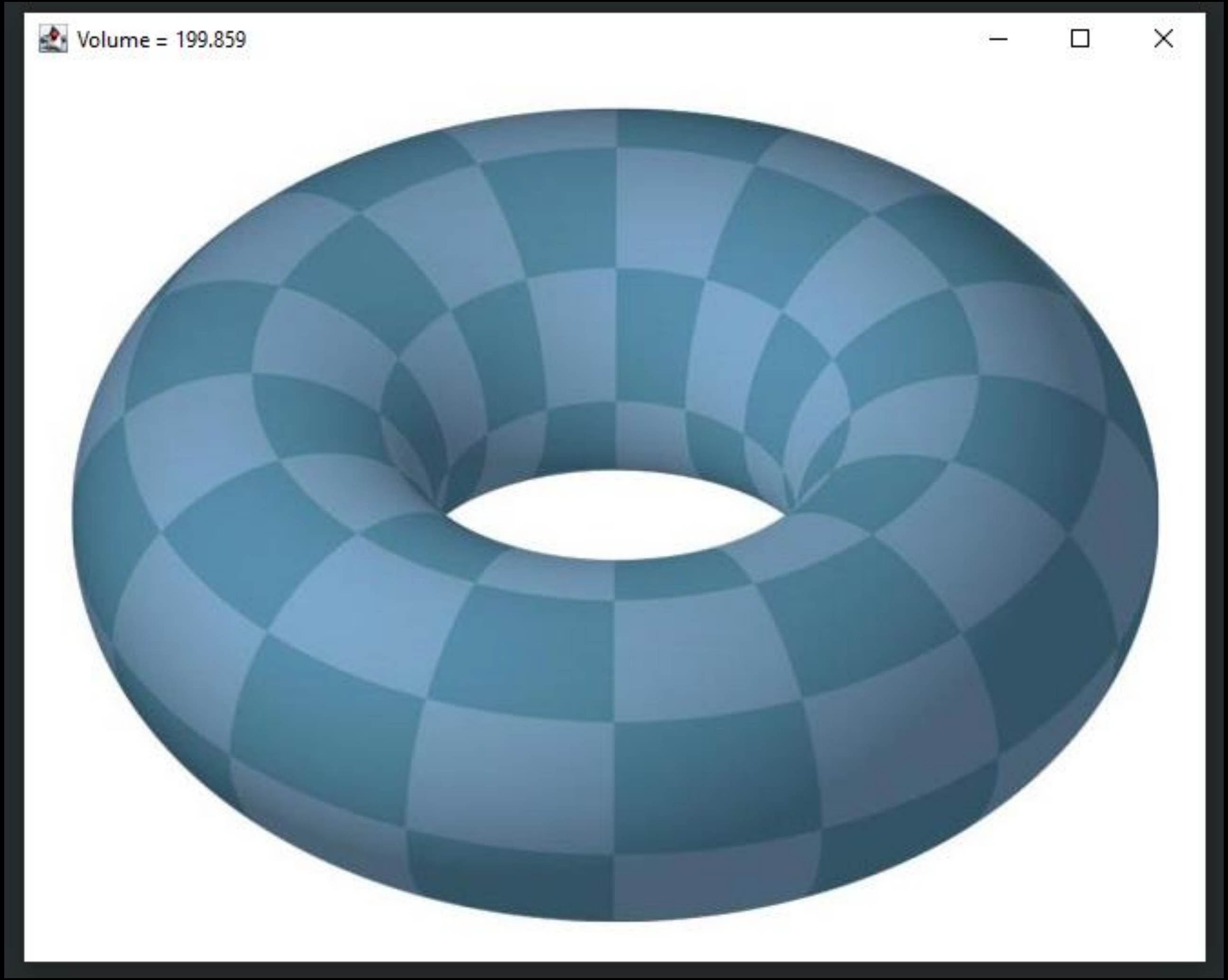
Input

Enter the outer radius of the Torus:

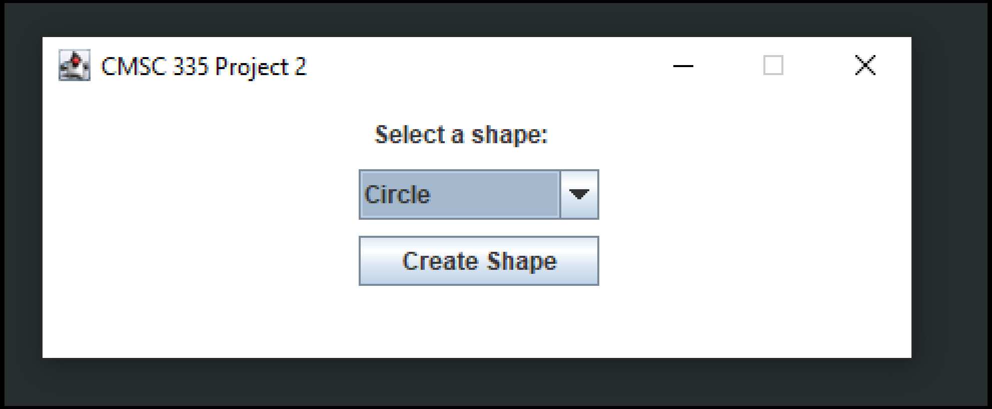
6

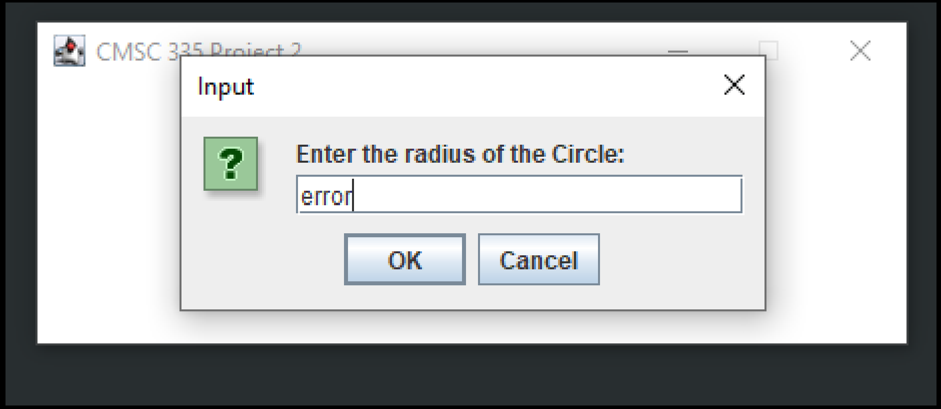
Construct a Torus:

Actual Output:
Volume = 199.859

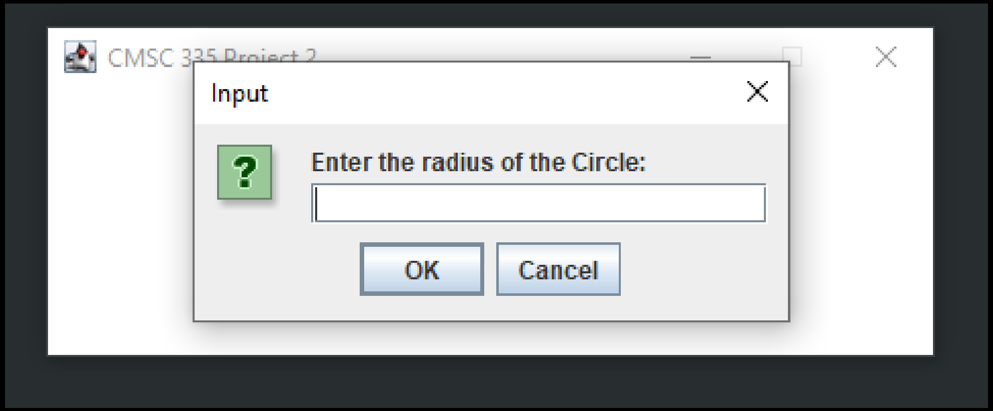

Volume = 199.859

Error Checking:

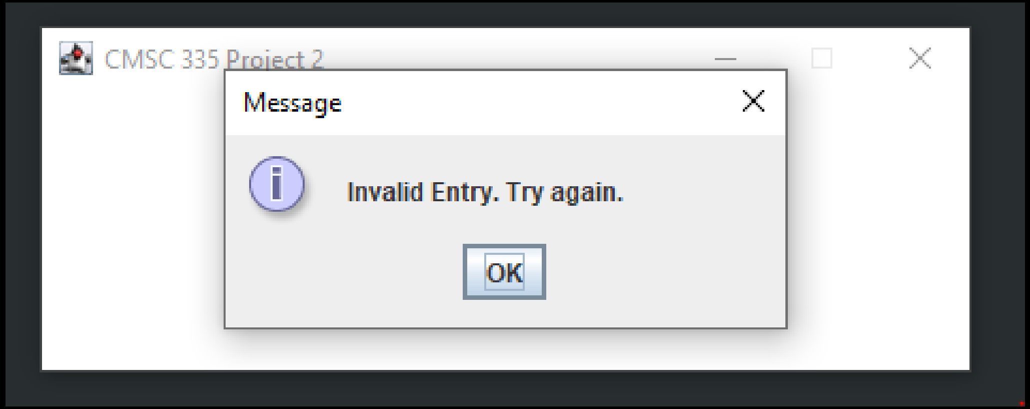




Enter the radius of the Circle:
error



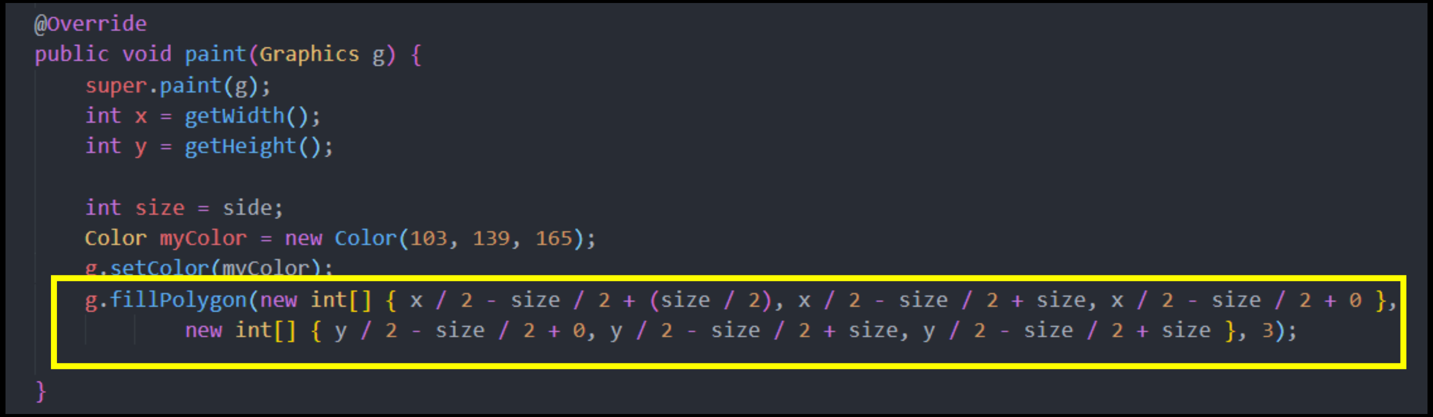
Enter the radius of the Circle:



Invalid Entry. Try again.

Lessons Learned:

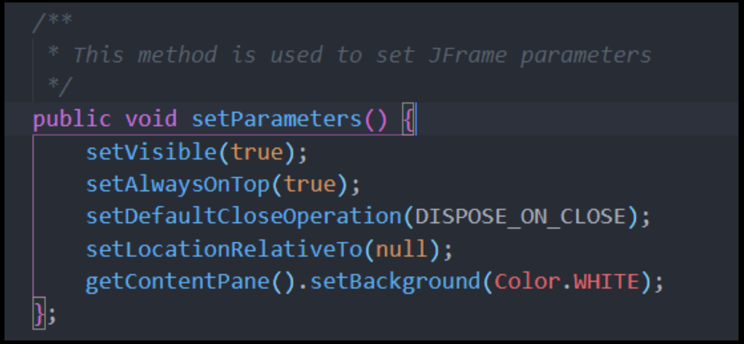
1. Dynamically centering the shapes on the JFrame, specifically a Triangle, was extremely complicated. It took many hours of research to learn how to do this. The final code is shown below:



```java
@Override
public void paint(Graphics g) {
    super.paint(g);
    int x = getWidth();
    int y = getHeight();

    int size = side;
    Color myColor = new Color(103, 139, 165);
    g.setColor(myColor);
    g.fillPolygon(new int[] { x / 2 - size / 2 + (size / 2), x / 2 - size / 2 + size, x / 2 - size / 2 + 0 },
            new int[] { y / 2 - size / 2 + 0, y / 2 - size / 2 + size, y / 2 - size / 2 + size }, 3);
}
```

2. During this project I noticed that I was repeating the JFrame code for each of the Shapes. To fix this issue, I leveraged the Parent methods, TwoDimensionalShape and ThreeDimensionalShape, and added setParameters() method. The final code is shown below:



```java
/**
 * This method is used to set JFrame parameters
 */
public void setParameters() {
    setVisible(true);
    setAlwaysOnTop(true);
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    setLocationRelativeTo(null);
    getContentPane().setBackground(Color.WHITE);
};
```