CSC 1350 Exam # 1

September 23, 2013

Section ①

NAME: ____________________________

- This is the hands-on programming part of your exam.
- **Do not open any application on your computer other than Netbeans.**
- This exam is an open book exam but no borrowing of textbook is allowed during the exam.
- Include header comments with the @author tag followed by your name before the definition of each class but no other Javadoc documentation is required.
- Read the instructions preceding each section carefully before beginning the section.
- When you are done, call the lab assistants and they will assist you and archiving and submitting your work using an electronic drop box that I have set up. Be sure to open the zip file and view its contents to ensure that Ellipse2DTester.java is included in it. **You will only be able to submit your work once so be sure that you are finished with the exam before attempting to submit your work.**
- Write your name and section on the exam sheet and turn it in to the lab assistant before you leave.

DURATION: 2 hours

Table 1: Distribution of Points

PART	WORTH	SCORE
Programming	60	/60

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO.

1 Programming

Instruction: Write Java Platform SE 7 compliant code for the programming exercise below. Use proper syntax and descriptive names for your identifiers but no other documentation is required. Be sure to import the relevant packages, where required.

Consider an abridged version of the Java Platform SE 7 API documentation for the *Ellipse2D.Double* class below:

java.awt.geom

Class Ellipse2D.Double

java.lang.Object

java.awt.geom.RectangularShape

java.awt.geom.Ellipse2D

java.awt.Ellipse2D.Double

Field Summary	
Modifier Type	**Field and Description**
double	height The overall height of the Ellipse2D.
double	width The overall width of this Ellipse2D.
double	x The X coordinate of the upper-left corner of the framing rectangle of this Ellipse2D.
double	y The Y coordinate of the upper-left corner of the framing rectangle of this Ellipse2D.

Constructor Summary
Constructor Description
Ellipse2D.Double() Constructs a new Ellipse2D, initialized to location (0, 0) and size (0, 0).
Ellipse2D.Double(double x, double y, double w, double h) Constructs and initializes an Ellipse2D from the specified coordinates.

Method Summary	
Modifier Type	**Method Description**
double	getHeight() Returns the height of the framing rectangle in double precision.
double	getWidth() Returns the width of the framing rectangle in double precision.
double	getX() Returns the X coordinate of the upper-left corner of the framing rectangle in double precision.
double	getY() Returns the Y coordinate of the upper-left corner of the framing rectangle in double precision.
void	setFrame(double x, double y, double w, double h) Sets the location and size of the framing rectangle of this Shape to the specified rectangular values.

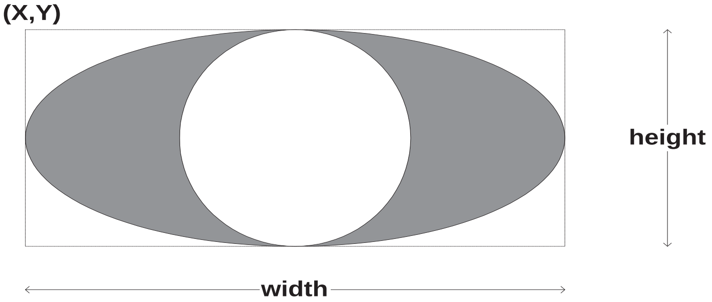

Figure 1: Initial Diagram of the Concentric Ellipses

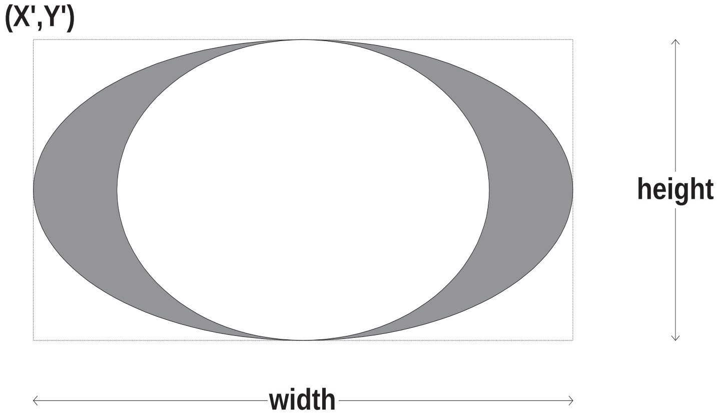

Figure 2: Final Diagram of the Concentric Ellipses

1.1 The *Ellipse2DTester* Class

The formula for the *area* of an ellipse is given by:

$$area = \frac{\pi.width.height}{4} \tag{1}$$

Write a program Ellipse2DTester that does the following:

1. creates a big ellipse using whose framing box has its upper-left corner located at (10,20) and whose width is 100 and height is 60,

2. creates a smaller ellipse (circle) whose center is the same as the bigger one and whose parameters are defined relative to the bigger ellipse as shown in figure 1. (Note that the x-coordinate of the framing box of the smaller ellipse would be

   $$x_s = x_b + \frac{width_b - height_b}{2}, \tag{2}$$

   where the 'b' and 's' subscripts denote the bigger and smaller ellipses, respectively. Also, note that the height and width of the smaller ellipse (circle) are the same.)

3. displays the location, width and height of each ellipse.

4. calculates and prints the area of the shaded region in figure 1.

5. shifts both ellipses rightward by 10 units and downward by 15 units.

6. changes the width of the smaller ellipse so that its width is 0.75 the width of the bigger ellipse.
7. again, shifts the smaller ellipse so that it continues to have the same center as the bigger ellipse as shown in figure 2. (Note that its new x-coordinate would be

   $$x_s = x_b + \frac{width_b - width_s}{2} \tag{3}$$

   .
8. again, displays the location, width and height of each ellipse.
9. calculates and prints the area of the shaded region in figure 2.

For each task above, call appropriate methods.
A typical program interaction will be:

```
Sample Run:

After creating the ellipses:
<-BIG ELLIPSE->
location: (10.0,20.0)
width: 100.0
height: 60.0
<-SMALL ELLIPSE->
location: (30.0,20.0)
width: 60.0
height: 60.0
Area of shaded region: 1884.9555921538758

After shifting and resizing the ellipses:
<-BIG ELLIPSE->
location: (20.0,35.0)
width: 100.0
height: 60.0
<-SMALL ELLIPSE->
location: (32.5,35.0)
width: 75.0
height: 60.0
Area of shaded region: 1178.09724509617240.
```