ASSIGNMENT 6 15 Points CIS 255

1. In **Chapter 14**, do the following exercises:
   Self-Review Exercises 14.1-14.3; Exercise 14.4-14.7.
   In **Chapter 15**, do the following exercises:
   Self-Review Exercises 15.1-15.3; Exercises 15.4-15.5.

   The above listed exercises are NOT to be handed in.

   For all programming assignments, your names MUST appear twice in your program: First, in the comments at the top of your code; and **second**, in your output. **NOTE**: the screensaver should **NOT** have a title bar so you cannot display your name and I will **NOT** deduct anything.

   At the top of every file this semester you MUST place the following:
   // Your Full Name
   // Your Email @my.smccd.edu
   // CIS 255 HJ
   // Class Name
   // Class Description
   // Assignment #
   // Date

2. **Use the shape classes from Assignment 5 for this program. This is a screensaver application that draws filled random squares, circles and a third custom shapes in random gradient colors using simple animation. The application will draw 100 random shapes, then clear the screen and draw 100 new random shapes.**

   Create a custom shape subclass (such as MyArc or MyTriangle). The custom shape must be fillable.

   Your DrawPanel class must be modified as follows:

   - **The DrawPanel class will implement MouseMotionListener. This means that you must implement the mouseDragged and mouseMoved methods. The mouseMoved method should call System.exit(0) so that the screensaver application will shut down. The mouseDragged method can just have an empty implementation { }.**

     Note: there is a small quirk with mouseMoved – on some Windows systems as soon as the screensaver is launched mouseMoved is triggered even if the mouse isn't moved. You should create a boolean instance variable firstTime initialized to true and in mouseMoved if firstTime was false only then would you call System.exit. Then you would set firstTime to false in mouseMoved so that any subsequent call to mouseMoved would be the result of actually moving the mouse rather than just starting up the screensaver application. The screensaver will then automatically shut down.

   The DrawPanel constructor should set the background color to black and add the MouseMotionListener.

- **The paintComponent method will create and draw ONLY ONE random filled shape – a rectangle, oval or custom shape – using a single MyShape reference to a MyRectangle, MyOval or custom shape object. There is NO loop in paintComponent.**
- **There should be a delay so that each random shape will appear separately. You can use this code, although you might want to change the argument to sleep:**
  ```
  try { Thread.sleep( 700 ); } catch ( Exception e ) {}
  ```
- **The first line of paintComponent will be an if statement that calls super.paintComponent if the number of shapes is 100. This will clear the previous 100 shapes so that you can start drawing the next 100 new random shapes. You should provide a counter as an instance variable so that you will know the number of shapes. Once it is 100 you should reset it.**
- **The last line of paintComponent will be repaint(); This will force paintComponent to execute again and draw the next random shape.**

Name your application ScreenSaver. This will be based on the TestDraw program from Assignment 5. Your screensaver should display without a title bar:

```
frame.setUndecorated( true );
```

Your screensaver must fill the entire screen. Do the following in main to get the screen width and height:

```
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
int width = screenSize.width;
int height = screenSize.height;
```

Then you can use these values to set the size of your application window. NOTE: you can call the getWidth and getHeight methods in the DrawPanel class to scale the random x and y coordinates.

You will need to set the background color also in main:

```
frame.setBackground( Color.BLACK );
```

You will submit all .java and .class files by WebAccess upload. You will need to submit the MyShape, MyBoundedShape, MyRectangle, MyOval, MyLine (if needed), and your custom shape files, as well as the DrawPanel and ScreenSaver files.

All shapes must be filled with a gradient paint. You need to generate a second random color and set the gradient paint in the paintComponent method. You should use a random boolean value so that the gradient paint will be either cyclic or acyclic. You will need to comment out the call to setColor in the draw method of each shape subclass so that the gradient paint will automatically fill the shape. There must be NO other modifications to the shape subclasses.

<u>SUBMIT</u>: You must submit a printed copy of all source code. All files must be compressed into a single zipped file named assign6.zip and uploaded to WebAccess. The zip file must be uploaded by 5:00 PM on the due date. Make sure that I receive ALL FILES, as I am not able to grade incomplete assignments.

An easy way to zip in Windows is to right-click on your file and then choose
Send To-> Compressed (zipped) Folder

On a Mac follow these instructions to zip your files:
http://www.macinstruct.com/node/159

Upload the assign6.zip file to the upload link in WebAccess by the due date/time.
http://smccd.mrooms.net

Example: