

CPSC-24500: Object-Oriented Programming

Homework Week 3

Name: _____

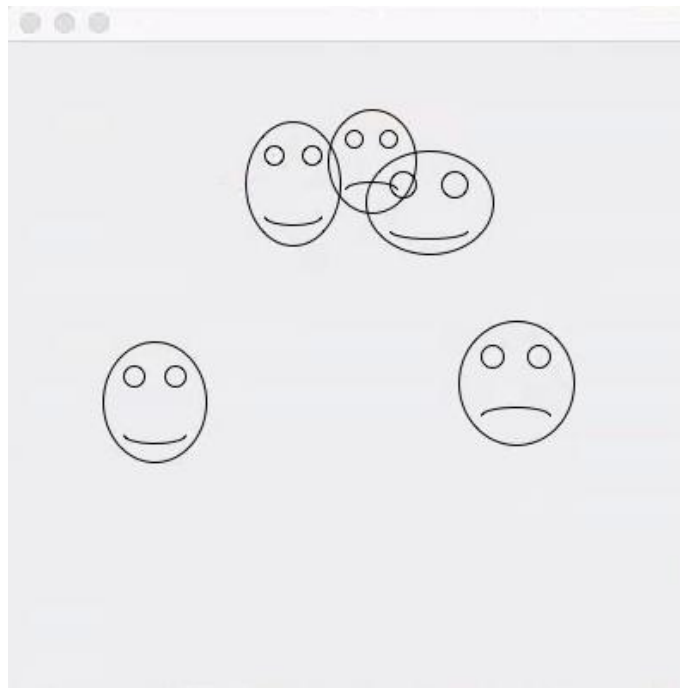
Score ____ / 30

Instructions: Fill in your name above and please answer the questions below. Submit your answers via the Blackboard homework submission link by the end of the day Sunday, March 26. This is an individual assignment and answers are not to be shared.

You will need to save a copy of the MS Word file to your local drive, fill out the name and date information, and answer each question by **highlighting** the best answer, by writing your answer after the question, or by inserting an image. Please let me know if you have difficulties.

1. Which of the following is a difference between ArrayList objects and arrays?
 - a. An array can store only one type of object, but an ArrayList list can store a variety of kinds of objects.
 - b. An array is fixed in size, whereas an ArrayList will automatically resize based on how many elements are stored in it.
 - c. An array cannot be used to support polymorphism, but an ArrayList can.
 - d. You don't have to create the array itself, whereas you do have to create the ArrayList object.
2. What is the job of a layout manager?
 - a. To render user interface components in a container.
 - b. To render a drawing on a canvas.
 - c. To position user interface components in a container.
 - d. To position a frame on a desktop.
3. The layout manager that positions controls based on north, south, east, west, and center sections is called
 - a. BorderLayout
 - b. GridLayout
 - c. FlowLayout
 - d. NullLayout
4. Which of the following is a difference between lightweight and heavyweight components.
 - a. Lightweight components render themselves using paint, but heavyweight components render themselves using paintComponent.
 - b. Lightweight components render themselves using paintComponent, but heavyweight components render themselves using paint.
 - c. Lightweight components can't contain other components, but heavyweight components can.
 - d. Lightweight components can't have layout managers, but heavyweight components can.
5. What is the purpose of calling super.paint(g) at the beginning of a JFrame's paint function?
 - a. It makes sure all the drawing tasks, like g.drawOval, are performed.
 - b. It makes sure that the Graphics object is created so that we can draw on it.
 - c. It makes sure that the background and borders and outlines of graphical user interface components are drawn.
 - d. It reduces memory leaks associated with drawing shapes on the screen.

6. Which of the following layout managers would be best for arranging the keys of a calculator?
- BorderLayout
 - GridLayout
 - FlowLayout
 - NullLayout
7. Which of the following represents the correct sequence of steps for how a JFrame is repainted?
- repaint triggers a call to paintComponent, which tells all the components owned by the frame to paint themselves.
 - paint triggers a call to repaint, which tells all the components owned by the frame to call their paintComponent function
 - repaint triggers a call to the paintComponent function of all of the components owned by the frame. The last component tells the frame to call its paint function
 - repaint tells the frame to call its paint function, which, in turn, calls the paintComponent function of each of the components owned by the frame.
8. Write the command needed to position a frame of width 400 and height 500 at 100 pixels from the left and 100 pixels from the top of the screen.
-
9. FaceDraw Assignment (22 points)



In this assignment, you will need to create an application in Java that draws random faces on a window. The application will draw 3 to 10 faces each time it is run. The faces will have a random width and height that are set to reasonable and visually appealing ranges. The application window size should be initially set to a reasonable size and all faces should draw entirely within the window. You do not have to worry about how to handle the resizing of the application window. Assume that it will remain the same size. Each face should have two eyes and a mouth. The mouth should be randomly smiling, frowning, or in-between.

#	Requirement	Points
1	The application must compile under the standard “javac” command line tools, run with the “java” runtime, depend on at most the (unmodified) ShapesLibrary.class file being located in the same folder, display a reasonably sized window, and display 3 to 10 faces at random.	6
2	The Face class is a model class. It (or its superclass) hierarchy must have, at a minimum, attributes (data members) that store its width, height, x, y, and smiling status. These attributes must all have setters and getters (even if they are not used in your application).	2
3	The Face class must have at least three constructors. One non-default construction should have all appropriate attributes passed into the method. The other non-default constructor should only have the application window height and width passed into the constructor and should set the other class attributes to appropriate random values.	2
4	The application must have view class called FaceFrame that Extends JFrame.	1
5	FaceFrame must contain an object called FacePanel that Extends JPanel.	1
6	The application main function must create an ArrayList called FaceList and populate it with between 3 and 10 random Faces... the 3 to 10 should be random and the Face attributes should be random as well.	2
7	The application main function should create a FaceFrame (which will create the FacePanel) and pass the FaceList into the FaceFrame constructor.	1
8	There are two possibilities for drawing the faces: (1) The FacePanel’s paintComponent function actually draws the Face objects where they are supposed to be and how they are supposed to look. OR (2) Implement a toGraphic(Graphic g) method in Face to draw (render) the Face.	3
9	Each time the face is drawn, the Face object’s toString() method should be called and the result should be displayed to the console window.	1
10	Provide appropriate comments.	1
11	Utilize at least one “assert” statement.	1
12	Submit your assignment as a single Java file named “Face_LastName.java” where “LastName” is replace with your last name.	1

If your solution does not compile and execute without errors when it is submitted, you will lose the 6 points AND I will send it back to you to fix and resubmit before I attempt to continue grading the assignment.

Do not copy another student’s work. I will use MOSS to detect plagiarism and will not ask for clarification if MOSS concludes you have copied another student’s work.

Tackle this problem gradually and make sure that you review the examples that we cover in class. The main goal of our discussions, lectures, and examples this week are intended to allow you to successfully deliver this application. Also, don’t hesitate to post something on our discussion board or to reach out to me directly if you need assistance.

I recommend creating the Face class first. Then, write your method that populates a list of Face objects with random faces. Then, create the window by extending JFrame, resulting in the FaceFrame class. Then, flesh out the FacePanel class. That last part will require calculating where to put the eyes and mouth and how to draw the mouth as smiling, frowning, or in-between.

Definitely pace yourself. Do not attempt to do this in one night.

Good luck – and have fun. This is, indeed, supposed to be fun.