CPSC-24500: Object-Oriented Programming

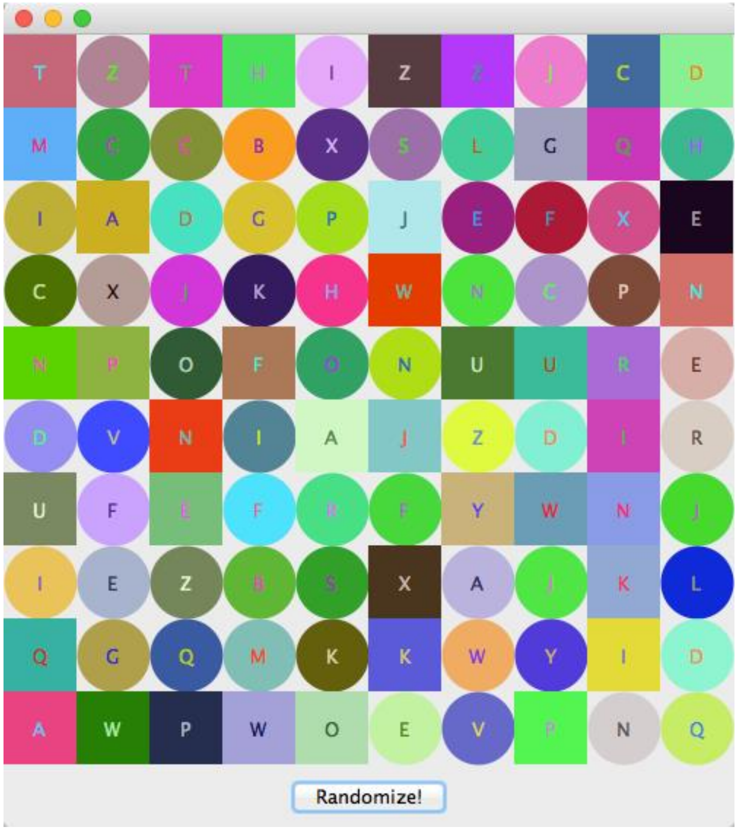
Homework Week 4 – Mosaic

Instructions: This week's assignment is solely a programming assignment named Mosaic. The assignment consists of a list of requirements and a detailed design. It will be your responsibility to implementing this design and for implementing one additional interesting feature of your choice.

This is the similar to how the real world works. Most often a non-technical person (project sponsor) will approach your group with a request. You or someone on your team will be responsible for collecting, documenting, confirming their requirements. How this occurs depends on your organization's development process (SDLC). For this assignment, I have collected your requirements and am providing your initial design.

Requirements: We would like to have a colored mosaic that shows a random pattern of colored circles and squares with a random letter in the middle of the shape.

Design: We will develop a Java application that provides a 10 by 10 grid of random circles and squares of random colors that includes a random letter in the middle. The resulting application should look something like:



Whenever the "Randomize!" button is pressed, the tiles will need to change at random.

At a minimum, your solution should include the following classes with the associated names:

Class	Description
Tile	Represents one of the shapes shown in the picture. Each tile can display itself as a colored circle or square. Each tile will have a color and a letter that it shows. Each tile will be placed in a particular row or column, so each tile object must keep track of those values, too. The Tile class must have a toString function that lists the various attributes of the tile, including its type, row, column, associated letter, and the red, green, and blue components of its color. Alternately, you may consider making a Tile descend from JPanel and utilize a grid layout control in your TilePanel (below) and insert a Tile instance into each grid item.
TileFrame	The heavyweight component that houses the panel of tiles as well as another panel that features the Randomize button.
TilePanel	The panel that occupies most of the TileFrame and paints and displays the set of tiles. Its paintComponent function will be responsible for rendering each tile in its proper place.
RandomizePanel	The panel occupies the "SOUTH" region of TileFrame and should include your RandomizeButton.
TileRandomizer	A class that has functions for building a tile with random parameters, for changing an existing tile's attributes to random values, and for changing the attributes of an entire array of tiles. You could call those functions buildTile, changeTile, and changeTiles, respectively. Alternately, you may consider making your Randomizer functionality part of your Tile class.

The Randomize Button could be a run-of-the-mill JButton, or it could be a JButton descendant. You must write an ActionListener either as an anonymous class or a named one that will randomize the tiles in the mosaic when the button is pressed.

The program will be graded as follows:

#	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 the 10 by 10 random grid upon startup. The name of your Java file and the name of your public method on the Java file will need to be "Mosaic" with a capital "M" only.	6
2	The Java file that you submit should start with a JavaDoc compatible comment that includes you as the Author (first name space last name capitalized appropriately). There should also be a comment describing the feature that you decided to add to the required solution.	1
3	The Tile class has all necessary parameters properly declared, at least two constructors, all necessary get and set functions, and a toString function that returns a comma-separated String that shows the tile's shape, row, column, letter, and color components.	5
4	At the beginning of each paint, the application should write "Start Paint***" once followed by writing the toString() of each Tile to console window utilizing System.out	2
5	The TileFrame class holds two panels, a TilePanel and Randomize panel that contains the "Randomize" button. It utilizes a BorderLayout to arrange them.	3
6	The TilePanel's paintComponent function draws all the tiles in the proper location with the proper colors and with the correct letters. The letters must be centered and rendered in a color that makes them visible.	4

7	When the Randomize button is clicked, the tiles are shuffled so that a new pattern emerges.	3
8	One feature of your choice. This feature should be commented succinctly as described in item #2 above. A couple ideas that come to mind include adding a menu that also provides executes randomize, implement a timer and have one in four Tiles update their letter every couple of seconds, submit the project through GitHub (you would need to work with me on getting this set up), or something else that catches your fancy. I may award points beyond four if the feature warrants.	4
10	Provide appropriate comments.	1
11	Submit your assignment as a single Java file named “Mosaic.java”. The file name must be the same as your public class that is run to execute the application. I will be copying the file to a folder, running “javac Mosaic.java”, and running “java Mosaic” to confirm that that you have completed step one successfully. You should include your full name in a comment at the beginning of the Java file that you submit.	1
	Total Points Possible	30

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.

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

Good luck – and enjoy!