# Lab 4 Fortune Teller GUI Program (20 pts)

## Learning Goals:

* Get practice with using the Java Swing GUI methods.
* Learn to use the online Java documentation to figure out how to do things with Swing that I have not shown you directly.

## Instructions:

**Create a java Swing GUI application that presents the user with a “fortune”.**

1. Create a java Swing GUI application in a new IntelliJ project called **FortuneTeller**.   
   Add your project to GitHub source control.
2. Your project will have a **FortuneTellerFrame.java** class (which inherits from JFrame) and a java main class: **FortuneTellerViewer.java**. Your application should have and use the following components:
   1. **Top panel:**  
        
      A JLabel with text “Fortune Teller” (or something similar!) and an ImageIcon. Find an appropriate non-commercial Fortune Teller image for your ImageIcon. (The JLabel has a constructor that takes a String and the ImageIcon. Figure out from the Swing API or the content in Canvas how to display the text either above or below the ImageIcon.) Select a font face that works with your image and set the size to a larger value (try 36 and 48). Note that you have to add the image file to your IntelliJ project directory.
   2. **Middle panel:**  
        
      A JTextArea within a JScrollPane where the fortunes will be displayed one per line. Again, set the font values so that it works. (Should be smaller than the large text of the Top panel.)
   3. **Bottom panel:**  
        
      A button with the label “Read My Fortune!”.

A button with the label “Quit”

1. Create separate fonts for the title, buttons and the fortune display.
2. The FortuneTellerFrame class inherits from JFrame. The minimal code in the java main class viewer just creates an instance of FortuneTellerFrame and sets the Frame stats.
3. Create a String arrayList<String> or String Array of at least 12 humorous fortunes. When the user clicks on the “Read My Fortune” button your program should randomly select a fortune from your array and **append** it to the display in the ScrollPanel TextArea. **Make sure that you implement the constraint that the program will never return the same fortune as the last one prese**nted**. To be clear, the program will repeat fortunes just not the same one twice in a row.** (Hint: remember the index of the current fortune and when you generate a new random index, make sure that it is different from the previous one.)  
     
   Note that the textArea will display all the fortunes that the user receives one after another and then will be scrollable.
4. Use a reasonable visually pleasing arrangement of your components using BorderLayout. Following the example that I have posted in Canvas, get an instance of the Toolkit and set your main JFrame to be ¾ of the width of the display and centered on the screen. (The example is in the Canvas Course Documents folder that contains the Java GUI materials.)
5. Make sure that you use the Java 8 Lambda Expressions for the actionlistener code for the buttons.
6. Test your program thoroughly. Get a screen capture shot that shows at least 5 fortunes in the scrollpane/textarea. **Paste the shot here:** A screenshot of a computer

   Description automatically generated
7. Be sure to commit and push all your files to the repo!
8. Rename this docx **LastnameFirstname \_Lab04.docx** (using your actual name) include your screen shots within as indicated. Submit this file and a URL link to your GitHub code repository.