CS: 114 **OPPDA** **Spring, 2023**

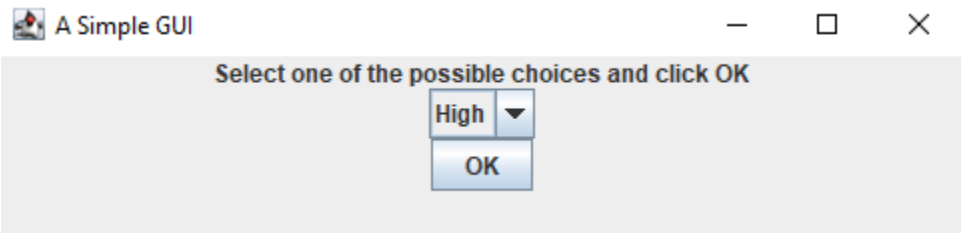
Lab5 (50 points) **Due 4/18/2023**

Submit your solutions to canvas. For programming assignments do not send the entire project. All that is needed are the files ending in .java. Each class will have to be defined in its own separate .java file. All driver code should be put in one .java file. Please make sure your name is included at the top of each .java file. Please put all files into one zip file and submit the zip file to canvas.

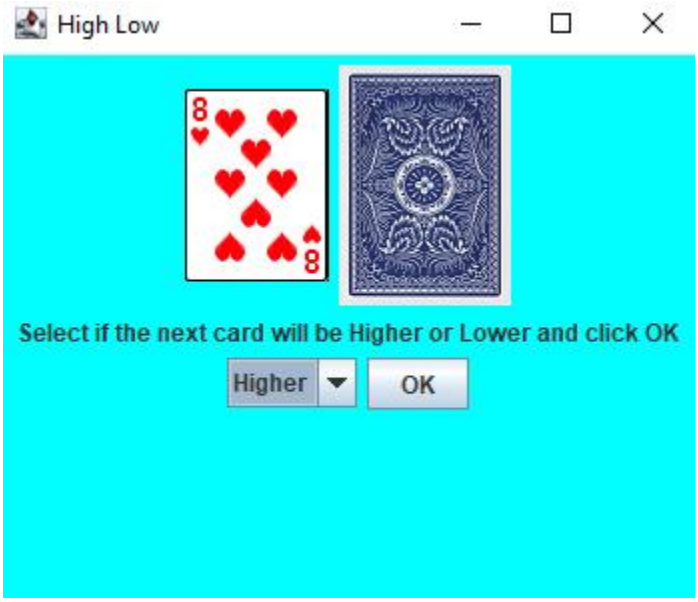
Goal: Practice controlling a program through the event loop

Problem 1

In this lab you are going to modify your solution from Project3 to run in a GUI event loop. You can use the code in `PullDownMenu.java` that was placed on canvas as a guide to adding the GUI your Project3 code.



Once your code integration is complete your window will look like:



You will not have to type anything in the console window to enter your guess. I will all be done via the GUI by selecting either Higher or Lower from the pull-down menu, and

then hitting the OK button. You will have to implement the `actionPerfomed` method using a lambda expression. `actionPerfomed` is the single abstract method in the Functional interface ActionListener. `actionPerfomed` accepts one argument of type `ActionEvent`. In this example you will not need to do anything with the event parameter. You lambda expression will have one parameter, call it e. The body of your lambda expression will simply call a method named `ProcessNextCard`. You will pass in the String that represents the guess selected via the GUI. The string will either be `"Higher"` or `"Lower"`. The `ProcessNextCard` method will look like:

```
public void ProcessNextCard(String guess)
{
    // Deal the next card and store it in the variable nextCard
    // Call the method DisplayNextCard(nextCard)
    // Check the prediction and output the result to the console screen
    // You will get extra credit if you can display the message to the window
    // Set the currentCard to nextCard;
    // Call the method DisplayCurrentCard(currentCard)
}
```

There should be no scanner inputs in `ProcessNextCard`. All input will be via the GUI.

You will add three new data methods to the `GHighLow` class. They are:

```
private DeckP3      deck;
private GCardP3     currentCard;
private GCardP3     nextCard;
```

In the `GHighLow` constructor you will build the three GUI items from `PullDownMenu.java`. They are:

```
JLabel label
JComboBox<String> comboBox
JButton button
```

You will add an `ActionListener` to the button with a lambda expression. After you create these three GUI components then do the following:

```
deck = new DeckP3();
deck.Shuffle();
// Get a card from the top of the deck
currentCard = deck.DealACard();
// Call method to display current card
DisplayCurrentCard(currentCard);

// this code already exists
this.frame.getContentPane().add(this.panel);
this.frame.pack();
```

```
this.frame.setVisible(true);
```

The driver will change to:

```
**
 * Graphical HighLow Game with GUI
 *
 * @author (Student Name)
 * @version (v1.0, 3/27/23)
 */
public class Lab5
{
   public static void main()
   {
       GHighLowP3 game = new GHighLowP3();
       //game.Play();
   }
}
```