

## CSC 161 SPRING 17 LAB 2-1 BORDERLAYOUT, GRIDLAYOUT, AND EVENT HANDLING

PROFESSOR GODFREY MUGANDA

### 1. LEARN TO GENERATE RANDOM NUMBERS

The class `Random` in Java can be used to create objects of the class `Random`, which in turn can be used generate random numbers. The `Random` class has a method

```
int nextInt(int b)
```

which returns a random integer  $x$  such that  $0 \leq x < b$ .

To make sure you understand how to use random numbers, write a Java program that prints out a sequence of 10 random integers that are less than 100.

### 2. OVERVIEW OF THE ASSIGNMENT

We are going to write a GUI application that allows a user to randomly generate an array of 10 integers in the range 0-99, and then click a button to have the program highlight the minimum number in yellow and the maximum number in pink (you can use any other two colors that you like, provided the numbers are still visible). The GUI will display the numbers in textfields of width 3.

### 3. CREATE AND DISPLAY THE TOP LEVEL GUI

Start by writing a program that displays a `JFrame` that will hold all user interface components. The `JFrame` should be  $400 \times 200$  pixels and be appropriately titled.



All of the code should be inside of the `main` method. Do not create a subclass of `JFrame`.

## 4. ADD AN ARRAY OF TEXT FIELDS AND TWO BUTTONS

Next, add the following local variables at the top of your `main()` method:

```
// MinMax and Randomize buttons
JButton randomizeButton = new JButton("Randomize");
JButton maxMinButton = new JButton("Max Min");

// Array to hold random numbers
final int SIZE = 10;
int [] rNumber = new int[SIZE];
JTextField [] tFs = new JTextField[rNumber.length];
```

Make sure to add a loop to actually create the arrays with width 3.

## 5. CREATE A SUBCLASS OF JOURNAL TO HOLD THE UI

Next, create a subclass `OuterPanel` of `JPanel` to hold the array of text fields and the two buttons. The class should have a constructor that takes three parameters, an array of `JTextField`, and two `JButton` objects.

```
class OuterPanel extends JPanel
{
    public OuterPanel(JTextField [] textFs, JButton b1, JButton b2)
    {
        // Set BorderLayout for this Panel

        // Add the topPanel with a GridLayout to hold the text Fields

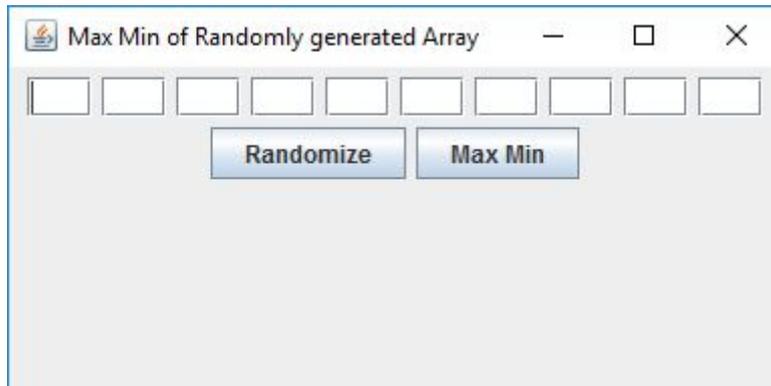
        // Add the bottom panel to hold the two buttons

        // Little margin to make things look good
        super.setBorder(new EmptyBorder(5, 5, 5, 5));
    }
}
```

The `JPanel` should have a `BorderLayout`.

- (1) In the NORTH region of the border layout, put a `JPanel` called `topPanel` with a `GridLayout` of one row as many columns as the number of elements in array of `JTextField` that was passed to the constructor. Add the text fields to `topPanel`.
- (2) In the SOUTH region of the border layout, put a `JPanel` called `bottomPanel`. Add the two buttons to `bottomPanel`.

Next, modify your `main` method so it creates an instance of the `OuterPanel` class, passing it the appropriate parameters. Add the instance of `OuterPanel` to the frame and show the frame as before. You should have something that looks like the following image.



#### 6. ADD A LISTENER FOR THE RANDOMIZE BUTTON

Begin by adding this statement near the top of the main method, to create a `Random` object, which you will use to generate random integers.

Next, add an action listener to the `Randomize` button. When this button is clicked, you will fill the `rNumber` arrays with randomly generated integers, and then set the text of the text fields with the `String` values of the random numbers like this

```
tFs[k].setText(String.valueOf(rNumber[k]));
```

Question: why does the simpler code

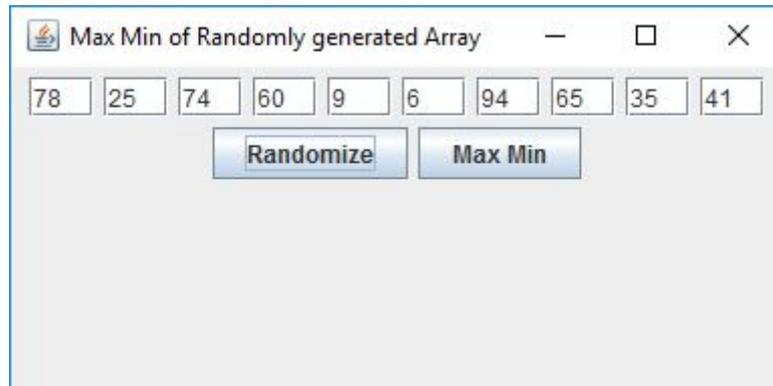
```
tFs[k].setText(rNumber[k]);
```

not work?

Put all the listener code after the code for creating the user interface, right after the call to make the frame visible.

```
// Add listener for randomize Button
Random randy = new Random();
ActionListener randomizeListener = evt ->
{
    // Code for event handler here
};
randomizeButton.addActionListener(randomizeListener);
```

Now clicking on the `Randomize` button should give a result as shown below.

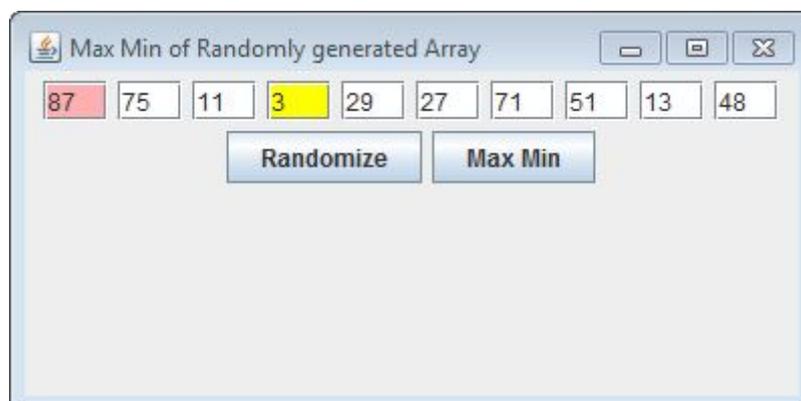
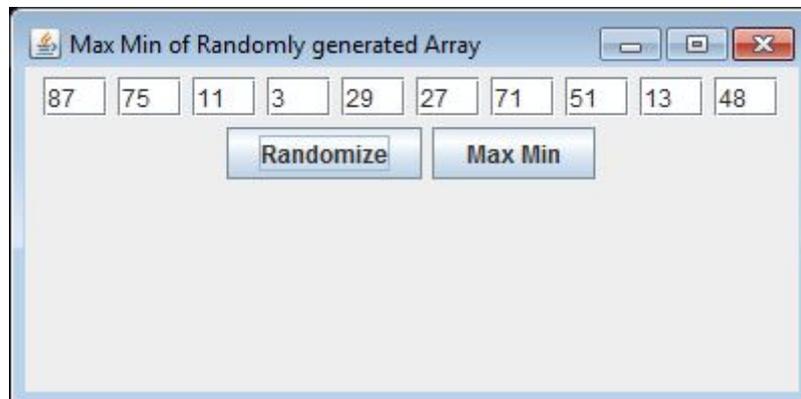


#### 7. ADD A LISTENER FOR THE MAX MIN BUTTON

Now for the final step. Add a listener for the Max Min button. Do this by adding two Color variables at the end of the code you have so far

```
// Add listener for maxmin button  
Color minColor = Color.YELLOW;  
Color maxColor = Color.PINK;
```

and then adding a listener for the Max Min button. Clicking the button should give a results as shown below.



8. DUE DATE

Wednesday of Week 2, at midnight.