

Lab Assessment 2: Stopwatch UI

The demonstration week for this assignment is the week of February (16-19th), 2021. **No demonstration will result in zero mark.** This lab assessment is worth 15marks (3% weightage).

Objectives

Continuing on what we developed in Lab Assessment 1 – Part 1-I (Stopwatch Application)

Tasks Involved

Having created the Frame, you'll now create the user interface by placing controls in the frame.

Follow the steps to continue with the Stopwatch example:

We'll place 9 controls in the frame: **three buttons, three labels and three text fields**. The buttons will start and stop the timing. The labels and text fields will be used to display the timing results: You must place these controls in a 3 x 3 grid as given below:

	gridx = 0	gridx = 1	gridx = 2
gridy = 0	startButton	startLabel	startTextField
gridy = 1	stopButton	stopLabel	stopTextField
gridy = 2	exitButton	elapsedLabel	elapsedTextField

- 1) Declare the nine controls
- 2) Replace the setSize line (from LA#1-Part1-I code) with the line establishing the grid layout:


```
getContentPane().setLayout(new GridBagLayout());
```
- 3) Set the properties of the nine controls:
 - a) startButton: text, gridx and gridy
 - b) stopButton : text, gridx, gridy
 - c) exitButton : text, gridx, gridy
 - d) startLabel : text, gridx, gridy
 - e) stopLabel : text, gridx, gridy
 - f) elapsedLabel : text, gridx, gridy
 - g) startTextField : text, columns, gridx, gridy
 - h) stopTextField : text, columns, gridx, gridy
 - i) elapsedTextField : text, columns, gridx, gridy

Notice how each control is positioned within the grid along with how you've set the number of columns for the text field controls. At this point, your interface has a finished look.

4) Add the control Listener:

In every project you'll construct, you need to "listen" for the event when the user closes the window. The adapter that implements events for the frame (window) is called the WindowAdapter and it works with the WindowListener. There are certain window events that can be "listened for." In our case, we want to listen for the windowClosing event. The code that adds this event method to our application is:

```
addWindowListener(new WindowAdapter()    {  
    public void windowClosing(WindowEvent e)  
    {  
        [Java code - when and how to close the window]  
    }  
});
```

For Swing components, like the button, label and text field used here, event methods (actionPerformed) are added using the ActionListener. If the component is named ctrlName, the method is added using:

```
ctrlName.addActionListener(new ActionListener()    {  
    public void actionPerformed(ActionEvent e)  
    {  
        [Java code to execute]  
    }  
});
```

The method has a single argument (ActionEvent e), which tells us about which particular event has occurred (each control can respond to a number of events).

- 5) Write the code for every event a response is needed for. In this application, there are 3 such events – clicking on each of the buttons.

Goals

To successfully complete this assessment and obtain all the marks, you must:

- Declare 3 class level variables under the lines declaring frame controls
- In the frame constructor, add the 'Window Closing' event method which tells the application when to stop. Similarly, create the event method for the startButton and stopButton after their respective constructor methods.
- User should not be allowed to click the stop button before the start button and once, start button has been clicked, user must click the stop button.
- Be able to compile and execute this code from the command line. You will be called upon to do so, so have your command prompt ready during your lab session.

Marking Scheme

Item	Penalty
UI components are not correctly arranged	2 marks
Controls got disabled/enabled Incorrectly	5 marks
Other specification violations	1 marks
Code violates Java formatting, variable naming conventions, comment headers	2 marks
Missed demonstration	15 marks
Ran in Netbeans, but not from the command line.	5 marks
Total	____ Out of 15.

Lab Deliverables

- You must submit your .java file in the assigned submission folder on BRS section 300 (activities → Assignments). We don't need your entire project folder, a java file is what we need.
- You will need to demo this code in your lab session, the week of February 16-19th, 2021. During demonstration, you must be able to answer any questions related to the assessment. For example: What needs to be done if you'd like to continuously display the end time and/or the elapsed time?