**Worksheet 13 – GUIs**

1. What are the 3 aspects of any Graphical User Interface?

Aspect 1: **UI Elements**

**These are the core visual elements the user eventually sees and interacts with**

Aspect 2: **Layouts**

**They define how UI elements hould be organized on the screen and provide a final look and feel to the GUI**

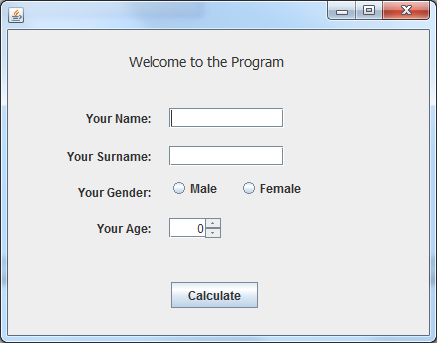
Aspect 3: **Behaviour**

**These are evenets which occur when the user interact with UI elements**

1. Write down the name of the different components that appear in every image below.

|  |  |
| --- | --- |
| Example | Component Name |
| Image result for java swing jframe | JFrame |
| Image result for java swing jslider | JSlider  JTextArea |
| Image result for java swing imageicon | JButton  JTextField |
| Image result for java swing jspinner | JLabel  JSpinner |
| Image result for java swing jmenu | JMenuBar  JMenu  JMenuItem  JSeperator |

1. Let’s write our first GUI Application. It is the same application that we had done as a console application (calculates user’s age in seconds).
2. Create a new Java Application called *FirstGUIProgram*.
3. Right click on the project, choose *New…JFrame Form*. Call it *MainScreen*.
4. Drag and drop components into the form to make it look like the following. Note that on the right you can see the available components as well as the active component’s properties. Also, on the left you should be able to see the Navigator, which shows the components that are in your form.



* Use the components’s text property to set (or empty) the text that is shown in the components.
* Also, for the components that will be used for user input (such as the textboxes), make sure you give them a meaningful name, by right-clicking, *Change Variable Name*. Try to use conventions, such as starting names of TextBoxes with *txt*, of RadioButtons with *rb*, etc…
* For the Male and Female radio buttons, you need to add them to a Button Group (so that only one of them can be checked at the same time). Also, set Male to be selected by default.
* For the age spinner, let’s set the min and max to 0 and 100. In the properties of the spinner, click on *Model…Spinner Model*. *Model Type…Number*, and set the min and max from there.

1. Let us run the program. Remember to run the correct class (the form). Now when running the program, we should see our window with the components.
2. We will now add some code such that the user is shown the result as soon as he clicks on the button. While in design mode (program not running), double click on the button.

The method that we are taken to is the event handler for the button click. Inside here we can put the code we want to run when the event happens.

In order to read the value from a TextBox we use:

String s = txtBoxName.getText();

In order to read the value from a JSpinner we use:

String s = spinnerName.getValue();

byte b = Byte.parseByte(s); //if we need to convert it to byte

int i = Integer.parseInt(s); //if we need to convert it to int

For the radio button, we can use the following code to know if the radio button is selected or not.

boolean selected = rbName.isSelected()

Work out the calculation and display a message box to the user, showing his particulars and the result. Use the following code:

JOptionPane.showMessageDialog(null, “what you want to tell the user”);

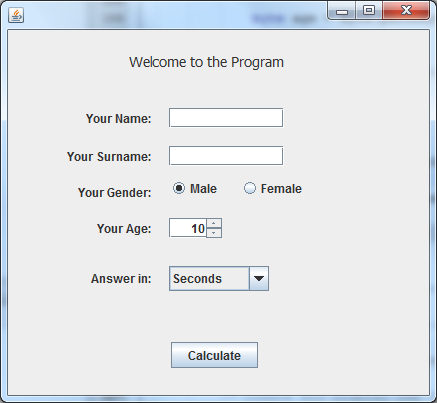
1. Let’s add some validation to our code. We want to show errors to the user when he leaves the values of Name or Surname empty. Start by adding invisible labels next to the two textboxes (set the font colour to red). Give them meaningful names.

Use if statements in the Button’s event handler to implement validation. Use the following code to change the text inside a JLabel.

labelName.setText(“New text”);

Remember to clear the error labels if the user’ input is accepted by the validation process.

1. Improve the program by adding a ComboBox in the GUI, using which the user can choose whether to get the results in seconds, minutes, or hours, as follows:



Use the following code to find out which index from the combo box the user has selected.

int i = cmbName.getSelectedIndex();