

Java Assignment#1

Java Programming



Name: _____

Mark: ____/100

Important Notes:

- Each team group (3 students) will work on this assignment individually. For example, team1 members can work together but they CANNOT work with or assist another member from a different team
- The assignment should be **submitted (online - GitHub Account) before the due date** in order not to lose marks. Each student must upload the assignment to their GitHub website (New repo as usual). The team coordinator will be in charge of submitting the links (URL) for their team members' GitHub on the assignment submission section of Canvas
- No need to think about adding a live demo with Java assignments :-), it's not front-end (HTML/CSS/JavaScript) development assignment and GitHub can only run static pages that build on front-end tools :-)
- Follow the **exact naming rules** for all your assignment files as explained in the instructions and shown in the diagram
- **At the end of this assignment**, you should have a **project folder named "Assignment1-Swing-Events" as an example or any other name you prefer as a team. This folder's name (Project Name) could be also the name of your GitHub repos where you have to upload your assignments**
- Use .gitignore to ignore all the other unnecessary folders:
 - .vscode/
 - bin/
 - lib/

(10 Marks)

Assignment Contents:

This assignment will be a demonstration of your skills in using:

- Java Classes, methods, formulas
- Java Class Inheritance
- Java Interfaces Implementation
- "ArrayList" class, "Arrays" class, and "List" interface
- Strings to "ArrayList"
- Swing GUI Interface
- AWT Events
- Essential Git Commands

You will be a Language Translator! But not from English to French 😊 You will translate the main logic of your code from JavaScript language to Java language.

Take the idea of the (JavaScript-Lab5) that you have finished in the AWP module (only the Excel part/page and ignore the user's form page) with the same functionality (input/output) and a similar GUI, and convert it to a JAVA desktop application using:

- Swing GUI (with any settings, layout, colours, ... as you prefer/like)
- AWT Events, for this task/assignment you will need to use (implement) the following Event Interfaces:
 - "ActionListener" Interface
 - "ItemListener" Interface
- String functions

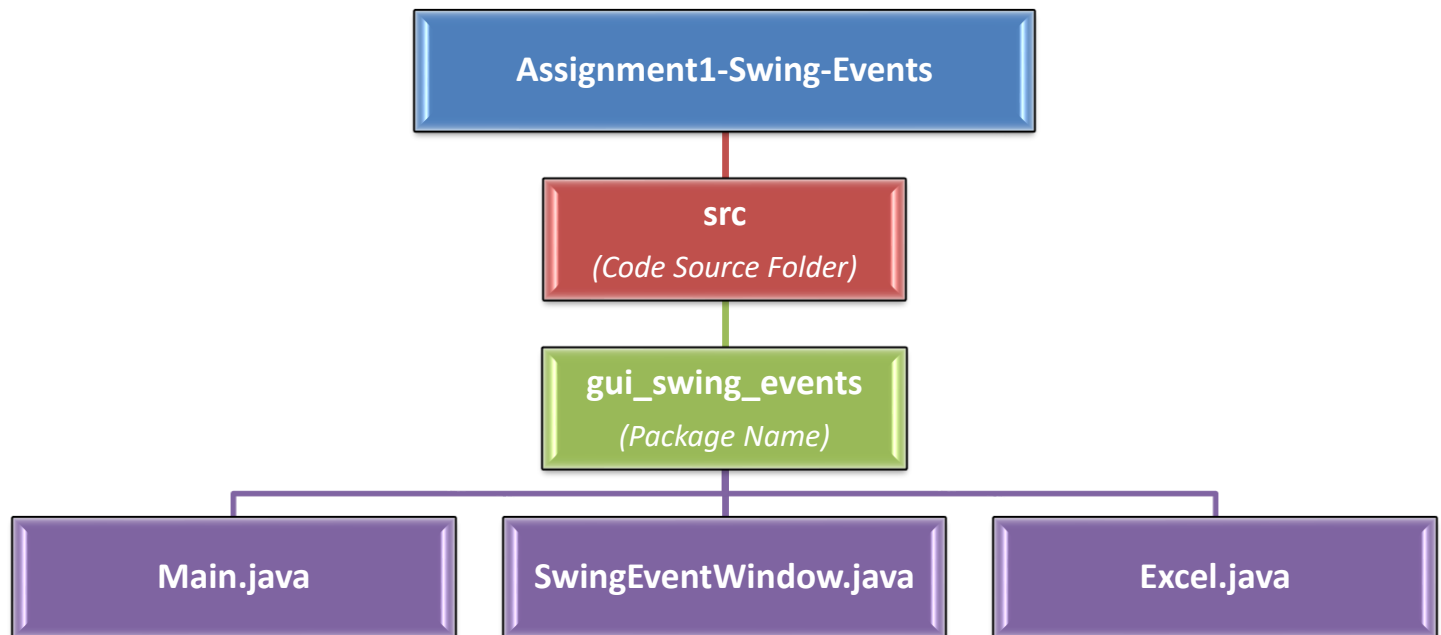
HINT: You can refer to the JavaScript assignment/task for the "Excel" page to guide through the main logic to follow in this assignment. However all the needed steps/logic are listed in this document.

Assignment Instructions:

1. Using any IDE/Editor you prefer, create a new Java project with the name "Assignment1-Swing-Events", "Swing-Events", or any project name you prefer
2. All your Java files should be inside a package named **"gui_swing_events"**
3. Create three Java class files:
 - a. **Main.java:**
 - i. The entry point to run your application → has the main() method
 - ii. Run the "SwingEventWindow" class constructors
 - b. **Excel.java:**
 - i. Contains the methods/solutions for calculating the 4 functions (Excel Functionality): (The total, The average, The maximum number, The minimum number)
 - ii. You will need to convert the string of numbers (user's input) into "ArrayList" of numeric values. Using ArrayList is more practical than using simple array (check the steps/the hints at the end)
 - c. **SwingEventWindow.java:**
 - i. Contains:
 1. Swing GUI,
 2. Events
 3. The calling of required Excel class method/function based on the user's input

You can check the sample image of the GUI interface in this document (it has the same fields as the AWP JS Task)

Assignment Folder Structure:



Assignment Pseudocode and Steps Guidelines:

Main.java

```
// the package statement

// the Class file that contains the main() method as the entry-point
// main() method for just establishing the "SwingEventWindow" class
```

(5 Marks)

Excel.java

```
// the package statement

// importing the required "java.util" packages:
// ArrayList class
// Arrays class
// List interface

public class Excel {
    // Declare the private class variable
    // This variable will be an empty "ArrayList" object of "Double" data type
    // To save/retrieve the user's input numbers in the textfield

    /*
    * Declare the first constructor method:
    * Passing an argument/parameter of ArrayList<Double>
    * Notice that this constructor will not be in used in this assignment,
    * but it's good to have it in case we need to immediately pass ArrayList
    */
    public Constructor1 {
        // Just a simple assignment operator
        // Assign the passing parameter it to the class variable "numbers"
    } // end Constructor1
```

```
/*
 * Declare the second constructor method (Overloading):
 * - passing an argument/parameter of String data type
 *
 * NOTE:
 * This passing string will represent the user's input (string of numbers)
 */
public Constructor2 {
    // NOTE: You can combine the three lines below into one:

    // Creating an array of String data type out of the string of numbers
    // using .split() method:
    // You can name it "strNumArray"
    strNumArray = ...

    // Creating a List view (List Interface) from the previous array
    // using .asList() method of the "Arrays" class
    // You can name it "strNumList"
    strNumList = ...

    // Creating an ArrayList object ("ArrayList" Class) from the previous List:
    // You can name it "strNumArrayList"
    strNumArrayList = ...

    // Finally, using forEach to loop through the previous "ArrayList" object
    // 1. Converting each item of String Data type to "Double"
    // 2. Adding the value to the "ArrayList" object "numbers"
    // NOTE: Using double so it can handle both whole numbers and fractions
    strNumArrayList.forEach(...);

    // for testing (optional) 😊
    for (int i = 0; i < numbers.size(); i++) {
        System.out.println(numbers.get(i));
    } // end for loop
} // end Constructor2
```

```
/*
 * Start declaring the 4 individual methods:
 * - findTotal() method
 * - findAvg() method
 * - findMax() method
 * - findMin() method
 *
 * NOTES:
 * - ALL the method should be publicly accessed
 * - ALL the method should return a double data type (for sure) :-)
 * - No need to pass any argument as you can call the ArrayList "numbers"
 * do your calculation on it
 */

// Define the first method "findTotal" to return the total
// Define the second method "findAvg" to return the average
// Define the third method "findMax" to return the maximum number
// Define the fourth method "findMin" to return the minimum number
} // end class file
```

(35 Marks)

SwingEventWindow.java

```
// the package statement

// import the required swing classes:
// Import the two required interfaces:
/*
 * Both interfaces have one required method (Event-Handler) to be implemented:
 * Each event-handler needs its event class to be imported also:
 * Import the two required event classes for each interface event-handler
 */

public class SwingEventWindow [ext. the GUI class, imp. 2 Interfaces] {
    // Declare the checked radio button variable flag:
    // this global and private variable is set to an initial value of 1
    // which refer to the first radio button to be selected by default
    // You can name "rdoChecked"
```

```
// Create JPanel main container object:
// You create JPanel sub containers objects for each section of the GUI:
// Create JComponents Items based on the assignment requirements:
// Create/Declare the 4 radio buttons objects:

// Using ButtonGroup class for grouping the 4 related radio buttons
// Create/Declare the ButtonGroup object
// for grouping the radio button inside the constructor method

// Class Constructor:
public Constructor() {
    // Setting Windows Title by targeting the method from the super class:
    // Setting Window (JFrame) Size:
    // Set the Window (JFrame) visibility to true to make it visible

    // Setting the default operation for the close button to Exit the JFrame
    // (Stopping the application)

    // No need to assign/set any value to the JComponents
    // It's already done when they are all declared above

    // Just group the four related JRadioButton components:
    // by adding them to the ButtonGroup object that you declared earlier

    // Adding the components to their panels:
    /*
    * IMPORTANT Note:
    * For quick demo, I used different and multiple JPanels objects
    * to layout the GUI components.
    * It's better to use the "GridLayout" based on the instructions
    */
    // Panel#1 for lblMain - Panel#2 for lblRequest - Panel#3 for txtNum
    // Panel#4 for the 4 radio buttons
    // Panel#5 for the calculate button
    // Panel#6 for the result label and the result output textfield

    // Adding all the 6 sub panels to the main panel (the main container):
    // Adding the main panel "mainPanel" (JPanel object) to the JFrame:
```

```
// Adding/attaching the required Events to some components:
/*
 * We need to add the required events to each radio button
 * and to the submit "Calculate" button only
 */
} // end constructor

// Implementing the two required methods (event-handler)
// for each interface:
@Override
public void Event-Handler method for the radio buttons {
    if (first radio button for total is selected) then:
        // set the value of the flag variable "rdoChecked" to 1

    else if (second radio button for average is selected) then:
        // set the value of the flag variable "rdoChecked" to 2

    else if (third radio button for max is selected) then:
        // set the value of the flag variable "rdoChecked" to 3

    else then:
        // set the value of the flag variable "rdoChecked" to 4
}

@Override
public void Event-Handler method for the calculate button {
    // Getting the user's input string from the first textfield:
    // Save it into a variable of String data type:

    // Initialize our "Excel" class object:
    /*
     * Passing the String value of the user's input to the constructor
     * Java will pick the second constructor from the Excel class
     * based on the passing data type of the argument which is "String"
     */
}
```

```
// Run the if condition for checking the value of the flag variable "rdoChecked"
if (rdoChecked equals 1) then:
    // call the findTotal() method
    // output/Print the result (value) in the result's textfield

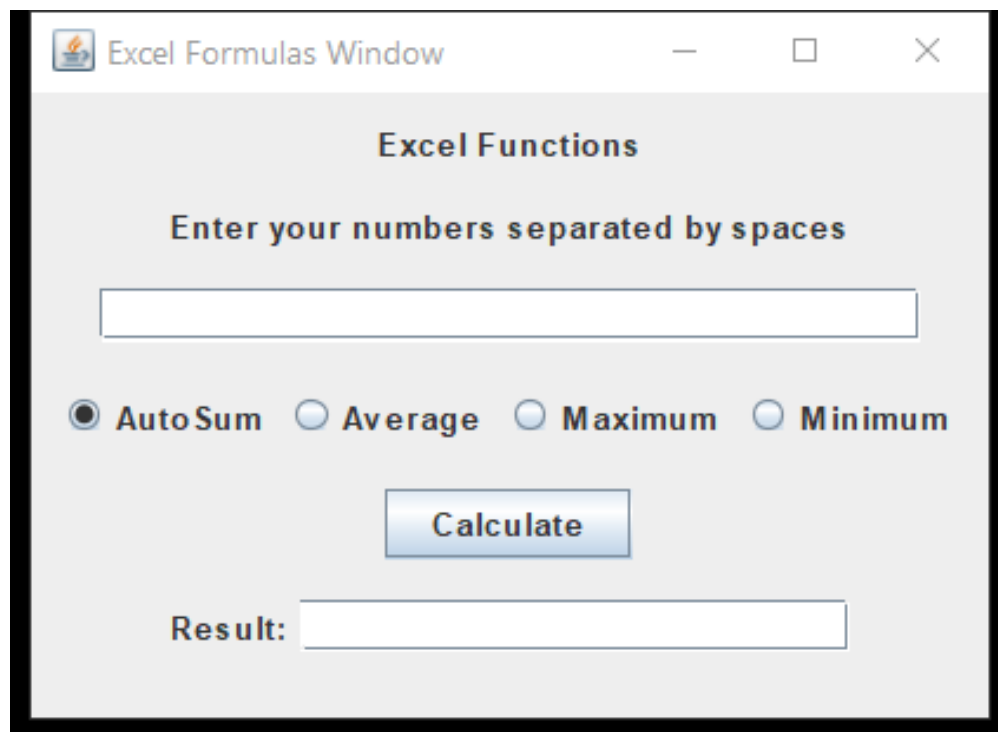
else if (rdoChecked equals 2) then:
    // call the findAvg() method
    // output/Print the result (value) in the result's textfield

else if (rdoChecked equals 3) then:
    // call the findMax() method
    // output/Print the result (value) in the result's textfield

else if (rdoChecked equals 4) then:
    // call the findMin() method
    // output/Print the result (value) in the result's textfield
} // end Event-Handler method for the calculate button
} // end class file
```

(50 Marks)

The Main Window Layout and Contents:



Hints and Tips:

For more review about the needed events, radio button groups, and converting Strings to “ArraList” object, you can check below my listed code samples on GitHub which are the lectures that we had covered about Collections, Swing, and Events as Guidelines:

- For Layout the element, you can use “GridLayout”:
 - [Grid Layout Example#1 Link](#)
 - [Grid Layout Example#2 Link](#)
- For modifying the appearance (Texts and Colours) plus Adding an event to the “Calculate” Button:
 - [Button Event Example1 Link](#)
 - [Button Event Example2 Link](#)
- For grouping the radio buttons and adding an event to each one plus read/write to/from the text fields:
 - [Input Dialog Box Example Link](#)
 - [Item Event Example Link](#)
- For reviewing the “ArrayList” class:
 - [My Array List Link](#)
- For converting the string of numbers to “ArrayList” of Double/Integer Data type:
 - [String to “ArrayList” conversion Link](#)

Happy Coding 😊