Advanced Programming 2021 – Year 2
Labwork 9: (This lab is worth 50 points out of 500 points for labwork this semester)

NOTE: ALL LABS TO BE COMPLETED IN PROJECTS USING ECLIPSE OR EQUIVALENT IDE (NO MORE TEXTPAD, EVER EVER!!!)

IMPORTANT NOTES:

- **NO COPYING PERMITTED AND ZERO MARKS WILL APPLY TO COPIED WORK. FURTHER ACTION MAY BE TAKEN AGAINST STUDENTS THAT HAVE BEEN FOUND TO COPY WORK.**

- **ASSESSMENT WILL INVOLVE ONE-TO-ONE QUESTIONS ABOUT YOUR SUBMITTED WORK. USE COMMENTS IN YOUR CODE TO ENSURE YOU DON'T FORGET WHY YOU WROTE CODE YOU MAY LATER BE ASKED ABOUT.**

- **ALL WORK MUST BE SUBMITTED TO MOODLE BY DATES SPECIFIED (2 LABS SUBMISSIONS OF FIVE LABS THROUGHOUT THE SEMESTER).**

- **MANY OF THE TASKS ASSIGNED BELOW CAN BE COMPLEX AND\OR THE DESCRIPTIONS MAY REQUIRE FURTHER CLARIFICATIONS. PLEASE USE THE AVAILABLE LAB TIMES TO ASK FOR CLARIFICATIONS AND ADVICE\HINTS ON THE TASKS BELOW.**

Part 1 – Using package keyword and Javadoc (15 points)

Create a project called **Lab9Part1**. Create a class called **StringUtility**. Place the **StringUtility** class in a package called **com.yourSurname.util** (where "yourSurname" is your actual surname, e.g., "raeside"). Add **two static methods** to the utility class called **getSumOfAcsiiValues(String s)** and **getProductOfAsciiValues(String s)**. The getSumOfAcsiiValues(String s) method will take each ascii value of a string passed and compute the sum of the ascii values (get each ascii value and add each together to return one integer, e.g., Luke = 76 + 117 + 107 + 101 = 401). The getProductOfAsciiValues(String s) method will take each ascii value of a string passed and compute the product of the ascii values (get each ascii value and multiply each together to return one integer, e.g., Luke = 76 * 117 * 107 * 101 = 96095844). Finally create a test class called **TestStringUtility**. Place this class in a package called **com.yourSurname.testUtil** and import and call the two utility methods of StringUtility using a sample string of your choice. Create a jarfile for the project and test that the Jarfile runs (set TestUtility class to be the main class for the jar).

Required activities and marking guideline:

- Created package structures (2 points)
- Create utility class with two static methods in package (6 points)
- Complete FULL Javadoc for class and methods (3 points)
- Test the utility method in a class in different package (2 points)
- Jar the project and run the jar file (2 points)

Part 2 – Resource bundles and internationalization (15 points)

Create an Eclipse Project called **Lab9Part2**. Create a simple JFrame class with one JLabel showing a single sentence, e.g., "This label shows a sentence". Add three buttons to the JFrame which displays the name of three languages of your choice. Change the label to translate the sentence shown in the label to each of the languages you have chosen using either the **ListResouceBundle** or **PropertyResourceBundle** approach (you may choose either approach!). All of the buttons and the label should change to appear in the current locale.

Required activities and marking guideline:

- Three translations classed for buttons and label (2 points each) (6 points)
- GUI constructed with listeners (3 points)
- Locales created and language switching active (4 points)
- Full translations working on buttons and label for 3 languages (2 points)

Part 3 – Anonymous objects and anonymous inner class (20 points)

Create an Eclipse Project called **Lab9Part4**. Create a JFrame called **InputFormFrame** with two inner classes called **RegisterPanel** and **SubmitResponder**. The **RegisterPanel** must extend JPanel and build a JPanel with a very simple input form (e.g. you can have one input field and\or label) and one button with the text set to "Submit". The second inner class called **SubmitResponder** must implement ActionListener and will be the responder to the submit button defined in the **RegisterPanel**. The **InputFormFrame** constructor (the outer class) will create and add a non-anonymous instance of the **RegisterPanel** (as a class field) and the **RegisterPanel** will add an anonymous instance of the **SubmitResponder** to the submit button. When the submit button is pressed in the **RegisterPanel** the input field will print the message "Submit completed" and the input field will be set as uneditable; also once the submit button is pushed the title of the outer **InputFormFrame** to "Submit Completed!!!" using setTitle.

[N.B.: Marks will not be awarded if the **RegisterPanel** and **SubmitResponder** are not **inner classes** of the **InputFormFrame**.]

Required activities and marking guideline:

- Outer Frame with non-anonymous instance of panel added (4 points)
- Inner RegisterPanel created to create simple panel (5 points)
- Inner RegisterPanel class adds anonymous SubResponder (5 points)
- SubmitResponder changes the outer frame title and panel field (6 points)