Advanced Programming 2021 – Year 2
Labwork 4: (*60 points out of 500 points for labwork this semester)
***For the next 5 weeks there will 60 marks per labwork to make up a 50 mark shortfall from 9 labs**

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 – Play a sound using Clip (15 points)

Create an Eclipse Project called **Lab4Part1**. Create a class called **PlayMySoundGUI** that will play a sound or loop a sound continuously from a file using the **Clip** interface (use a .wav file that will play for a few seconds, e.g., the 1-welcome.wav file supplied in Moodle). The GUI should have THREE buttons to play, loop (continuously), and stop the sound. Use the AudioSystem and AudioInputStream classes to import the audio as described in the lecture.

Required activities and marking guideline:

- Create the GUI with the three buttons (2 points)
- Create the application code to access the sound (project directory) (2 points)
- Test the play button on the sound (listeners) (3 points)
- Test the loop button on the sound (listeners) (3 points)
- Test the stop button on the sound (listeners) (3 points)
- Test the application makes the sound (2 points)

Part 2 – Using Locales to display local sensitive information (15 points)

Create an Eclipse Project called **Lab4Part2**. Create an internationalized application called **PrintInternationalInformation** that will print out specific information in a regional format, as specified below. The application will simply create THREE Locale objects of your choice (e.g., French for France, English for the UK, German for Germany) and print out three locale-sensitive pieces of information in EACH of the locales you have created. The three pieces of information can be printed out using **System.out** and the information you should print in a localized format are: Today's Date (the date when you did the lab in LONG form), The days of the weeks and the cost of a dozen eggs (just use a price from the local supermarket and include the currency).

Required activities and marking guideline:

- Created THREE Locale objects (3 points)
- Display todays date in EACH of the Locales (4 points)
- Display the days of the week in EACH of the Locales (4 points)
- Display the cost of a dozen eggs (e.g. 2 euro) in the local currency (4 points)

Part 3 – Basic Internationalized class (ListResourceBundle) (10 points)

Create an Eclipse Project called **Lab4Part3**. Create a JFrame application called **Translator** that contains three labels with the strings "Nine", "Eight", "Seven" and a button with "Translate to Spanish" displayed on the button text. The English text must come from a **PropertyResourceBundle** for English (_en) (i.e., use the getString() method to show the text instead of hard-coding the text). Use the **PropertyResourceBundle** approach to translate the label texts to Spanish, i.e., "Nueve", "Ocho" and "Siete" when the "Translate to Spanish" button is pushed. Translation to Spanish MUST also be achieved by coding a Spanaish **PropertyResourceBundle** (e.g. _es file).Hard-coding of the strings without using the resource getString() will not receive any marks. For this task there is no need to re-translate back to English again.

Required activities and marking guideline:

- Implement the simple GUI with listeners (3 points)
- Implement the PropertyResourceBundle code for Spanish and English (5 points)
- Translation to German button works to change button text (2 points)

Part 4 Internationalized GUI – Translation Two Ways (20 points)

Create an Eclipse Project called **Lab4Part4**. Create an internationanlized JFrame application called **FullTranslationGUI** that contains ONE JButton with the label "List All Locales" and ONE JTextArea and a JComboBox (use the Calendar.getAvailableLocales( ) to get the list of available locales as shown in the lecture). Use an EXTERNAL PROPERTY FILE to carry out the translations of the internationalized text (use **ResourceBundle** class directly (easiest way to achieve this!) or use **PropertyResouceBundle**). Output ALL available Locales to the JTextArea when the JButton is pushed: NOTE: The list of locales must also be localized, i.e., the list of locales must be printed out in the language currently chosen. Set the JComboBox text to list the options for "English" and the second language (e.g. German, French etc.). Use Resource bundles to translate the English JButton to the other language of your choice (e.g. "Montrer tous les Locales" when the French option is selected). The JComboBox must also be internationalized (e.g. when displaying French the English language is displayed as "Anglais") . Finally each time the GUI language is changed the GUI must be capable of switching completely between languages continuously.

Required activities and marking guideline:

- Build the basic GUI button, combo and text area (3 points)
- Listeners for buttons and combo (3 points)
- External property files written (English and French) (6 points)
- Translation of ALL components to other language works correctly (4 points)
- Translation of ALL components back to English works correctly (4 points)