Advanced Programming 2015 – Year 2
Labwork 4: (5% - or 50 points out of 500 points for labwork this semester)

NOTE: ALL LABS TO BE COMPLETED IN PROJECTS USING ECLIPSE

NOTE: YOU MUST USE YOUR OWN EXAMPLES FOR THESE EXERCISES, I.E., YOU CANNOT RE-USE LECTURE EXAMPLE(S) AS SUBMISSIONS (THESE WILL BE GIVEN ZERO MARKS\POINTS)

Part 1 – Play a sound in a simple GUI (8 points)

Create an Eclipse Project called **Lab4Part1**. Create a class called **PlayMySoundApplication** that will play a sound from a file (use a simple sound like .au or .wav) using the **AudioClip** interface. You will need some headphones to test this in the lab if you are using the lab PCs.

Required activities and marking guideline:

- Create the application code to access the sound (3 points)
- Call play on the sound (AudioClip) (3 points)
- Test the application makes the sound (2 points)

Part 2 – Using Locales to display local sensitive information (12 points)

Create an Eclipse Project called **Lab4Part2**. Create an internationalized application called **PrintInternationalData** 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: The cost of a pint-of-milk (you may assume 2.5 units in whatever currency but you must print it in the actual currency for the locale if available). Javadoc and jar the project.

Required activities and marking guideline:

- Created THREE Locale objects (3 points)
- Display todays date in EACH of the Locales (3 points)
- Display the days of the week in EACH of the Locales (3 points)
- Display the cost of a pint-of-milk (use 2.50) in the currency of EACH of the Locale currency (3 points)
- Javadoc and Jar the project so that it runs (1 point)

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

Create an Eclipse Project called **Lab4Part3**. Create a JFrame application called **ButtonTranslator** that contains four buttons with the strings “One”, “Two”, “Three” and “Translate to French” displayed on the button text. The text must come from a **ListResourceBundle** class for English (_en) (i.e., use the getString() method to show the text instead of hard-coding the text). Use the **ListResourceBundle** approach to translate the button texts to French, i.e., “Un”, “Deux” and “Trois” when the “Translate to French” button is pushed. Translation to French MUST also be achieved by coding a French **ListResourceBundle** subclass (_fr), 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 four button GUI with listeners (3 points)
- Implement the ListResourceBundle classes for French and English (5 points)
- Translation to French in button works to change buttons (2 points)

Part 4 Internationalized GUI – External Resource File (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” (use the Calendar.getAvailableLocales( ) as shown in the lecture) and ONE JTextArea and a JComboBox. Use an <u>EXTERNAL PROPERTY FILE</u> (**PropertyResourceBundle** class, the property file itself will need to be placed in the **bin** directory using the code approach used in the lecture) to carry out the translations of the internationalized text. Output ALL available Locales to the JTextArea when the JButton is pushed: <u>NOTE: The list of locales must also be localized, i.e., the list of locales must be printed out in the language chosen</u>. Set the JComboBox text to list the options for “English” and “French”. Use Resource bundles to translate the English JButton to French “Montrer tous les Locales” when the French option is selected. The JComboBox must also be internationalized to read “Francais” and “Anglais” when French is the selected language. Finally when the JComboBox is displaying French, selecting the translated option “Anglais” must turn ALL components back to English again.

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 PropertyResourceBundle class written (4 points)
- External property files written (English and French) (4 points)
- Translation of ALL components to French works correctly (3 points)
- Translation of ALL components back to English works correctly (3 points)