GUI Programming 2024 to 2025 – Year 2

Labwork 3: (10% overall or 100 points of 600 points for labwork this semester)

Topic: Beginning SWING: Fields, JComboBox, JLists, RadioButton, Layout Managers

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. A COMPLETED SELF-ASSESSMENT SHEET WILL BE USED TO GUIDE THE ASSESSMENT. 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 (SUBMISSION DEADLINES WILL BE POSTED ON MOODLE).**

- **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.**

- **YOU CAN USE A SIMPLE JAVA ENABLED TEXT EDITOR IF YOU WISH, e.g., TEXTPAD or NOTEPAD. HOWEVER, I SUPPORT THE MOVING ON TO A MORE ADVANCED IDE AT THIS POINT ALSO (e.g., Eclipse or IntelliJ or NetBeans).**

- **CHATGPT and other similar AI tools that can code simple solutions are not permitted. THEY DO NOT TEACH YOU HOW TO BECOME A GOOD PROGRAMMER EITHER!**

Part 1 – Buttons with a listener (20 points)

a) Create a class called **Lab3Part1a**. Create a JFrame that contains a JLabel with the text "My name is X" and a JButton with the text "Translate to Y" (where X is your full name and Y is a language of your choice e.g., could become "Joe Bloggs is aimn dom" in Irish). Both components should be in a JPanel. Implement the action listener and corresponding event handler so that when the button is pressed the text in the label changes stating in the chosen language what your full name is.

- Create the Label and button and add to panel and frame (2 points)
- Add the panel to the Frame (remember content pane!) (1 point)
- Add listeners to the frame and button (3 points)
- Implement the handler method (actionPerformed) (2 points)
- Set the new text of the label when button pushed (test it!) (2 points)

b) Create a class called **Lab3Part1b**. Create a JFrame that contains a JButton with the text "My Text Changes When You Push Me". Add the button to a JPanel. Implement the action listener and corresponding event handler so that when the button is pressed the text in the button changes to "See, I Told You The Text Would Change?".

- Create the button and add to panel and frame (2 points)
- Add the panel to the Frame (remember content pane!) (1 point)
- Add listeners to the frame and button (3 points)
- Implement the handler method (actionPerformed) (2 points)
- Set the new text of the button pushed (test it!) (2 points)

Part 2 – JList with listener (20 points)

Create a Java program called **Lab3Part2**. Add a JList to the top of the JFrame that lists three types of animals (or sports, or anything that you can represent with an image: you can even mix and match things, e.g,. animals and sports). Add a JLabel to the center of the JFrame that will display an image: initially you can set a message in the JLabel that states "Images will appear here: please select from the above list". When the user selects an item from the JList make an image of that item appear in the label. Add a JTextArea to the bottom of the JFrame that says something about the image shown, e.g., if it's an image of a Tiger the text area could state: "This is a Sumatran Tiger – largest cat in the world". Use panels to contain the components.

- Create the Label and list and add to panel and frame (3 points)
- Add items to the JList (using an array is easiest) (2 points)
- Add listeners to the list (4 points)
- Make the switch of the images work using handler (3 images min) (6 points)
- Make the JTextArea display the correct information for the images (5 points)

Part 3 – JComboBox with listeners (30 points)

Create a JFrame class called **Lab3Part3**. Create a JFrame with TWO **JComboBox's** at the top of the JFrame. The first combo box will contain a list of colours. The second combo box will contain a list of images to do with your life, e.g., it could be sporting achievements, sports team, your favorite possessions, your family, your pet, images of books you've read….anything that is appropriate but unique to you. There should be at least five things you include in the second combo box, e.g., the combo box could contain a list of: "Favourite Team", "Dream Car", "Dream home", "Dream Job", "My home town". Add a JLabel to the center of the frame (place the label in a panel) that will display images when the combo box of favourite things is selected, e.g., selecting "Dream Home" in the Combo Box of favourite things will display an image of your "Dream Home" in the label. Add a second label to the bottom of the JFrame and set it to show the text "This label changes colour based on what you select". Place this label in a panel. Make the label change colour to the colour selected in the colour changing combo box (the background colour must change NOT the text colour!!). (*Note:* You may have to use a method called *setOpaque(true)* on the JLabel to get it to show the change of colour)

- Create the TWO JComboBox's with the items listed (8 points)
- Create and add the JLabel in the right location (bottom of frame) (3 points)
- Set initial text in label (1 point)
- Use panel(s) (2 points)
- Add the ActionListener to both of the combo boxes (4 points)
- Get the favourite things to show up in the centered label (6 points)
- Get the label at the bottom of the interface to change colours (6 points)

Part 4 – First full working GUI – Mobile top-up (30 points)

Create a class called **Lab3Part4**. Create a JFrame to model a simple Mobile top-up system. The GUI should provide a Top-up button, a Make-call button and a Send-text button that will reduce or add to the balance of the phone (you can use your own pricing system). At all times the balance should be visible on the screen using a label and the balance should not be allowed go below zero. It is up to you to design the layout of the GUI, marks will be awarded for the quality of the 'look' of the GUI.

- Use of colours in the GUI (3 points)
- Use of Font's and good labelling (3 points)
- Use of Layout's to make the GUI well structured (4 points)
- Add labels to instruct and guide the user (2 points)
- Implementation of the event listeners and handlers (all buttons) (6 points)
- Send-text reduces the balance and updates correctly (3 points)
- Make-call reduces the balance and updates displayed balance (3 points)
- Top-up values works correctly (2 points)
- System fully working (e.g. zero balance deal with) (4 points)