**GUI Programming 2015 – Year 2**

**Labwork 8: (5% - or 50 points out of 500 points for labwork this semester)**

**NOTE: ALL LABS TO BE COMPLETED USING TEXTPAD. NO COPYING PERMITTED AND ZERO MARKS WILL APPLY TO COPIED WORK.**

**ALL WORK SUBMITTED TO MOODLE BY DATE SPECIFIED (2 LABS SUBMISSIONS OF FIVE LABS THROUGHOUT THE SEMESTER)**

**Part 1 – JFileChooser (10 points)**

Create a class called **Lab8Part1**. Create a JFrame that contains a label called *fileInfoLabel* and a JButton with the text “Load file” (you can either use a panel or use different regions of the BorderLayout to organize the components). Code the JFrame so that when the button is pushed a **JFileChooser** is launched and the name of the file selected use the chooser is printed to the *fileInfoLabel.*

* Create the label and add to JFrame (1 point)
* Create the button and add the listeners (2 points)
* Organize the components (e.g. in panel or using layout) (1 point)
* Launch the JFileChooser (to select\open file) (2 points)
* Retrieve the file information from the JFileChooser (2 points)
* Update the label to show the file name selected (2 points)

**Part 2 – JColorChooser (10 points)**

Create a Java program called **Lab8Part2**. Create a JFrame that contains a label called *colourFontLabel* and set the text to “This text will change colour”. Use a large font in the label (e.g. minimum 25 and bold). Include a button with the text set to “Change text colour”. Implement listeners so that when the button is pushed a **JColorChooser** is launched. Set the text in the label to the colour that is chosen by the user using the colour chooser.

* Create the label and add to JFrame (1 point)
* Create the button and add the listeners (2 points)
* Set the font of the label (1 point)
* Launch the JColorChooser (to select colour) (2 points)
* Retrieve the colour information from the JColorChooser (2 points)
* Update the label to change the colour of the text shown in label (2 points)

**Part 3 – MouseInputListener (10 points)**

Create a JFrame class called **Lab8Part3**. Create a JFrame that contains one label in a panel to output mouse event data. Implement the **MouseInputListener** at the class level. Add the **MouseInputListener** to the panel and use the label to output the name of **ALL** mouse events detected (seven event handlers), e.g., “Mouse Clicked Event Detected” etc.

* Implement the MouseInputListener at the class level (1 point)
* Add the mouse listeners to the panel (2 points)
* Implement **ALL** handler methods so they display info in label (7 points)

**Part 4 – An application using mouse events and choosers (20 points)**

Create a class called **Lab8Part4**. Create a JFrame that contains four evenly split panels with a light black border (see examples for setting borders). Add a JLabel called *imageLabel* to the first panel. When the first panel is clicked by the mouse make a **JFileChooser** appear. Allow the user to pick an image to display in the label of the first panel and display the chosen image in the panel (the image should fit in the panel area, i.e., not too big!). In the second panel when the mouse enters the panel launch a **JColorChooser**, the user must choose a colour and the background of that panel should change colour. Add a label called *locationLabel* to the third panel. In the third panel listen for mouse **drag** events. When a mouse drag event is detected print the x and y co-ordinates of the event to the label called *locationLabel*. Finally, add a label to the fourth panel called *infoLabel*. When the mouse **exits** the fourth label the label should display the text “Bye bye mouse!!!”.

* First panel to change image in panel using JFileChooser (5 points)
* Second panel to change colour with JColorChooser (5 points)
* Third label to display x and y location of drag events (5 points)
* Fourth label to display goodbye mouse message on mouse exit (5 points)