EECS1720
Worksheet 5 – Java AWT/Swing GUI's

SOLUTIONS

1) Explain how a basic JavaAWT/Swing application is constructed (what are its main components)?

- Can either **aggregate** a Frame/JFrame object (use in a main or as a class field), then instantiate when the class is constructed, and invoke methods to set its size/visibility.
- OR can **inherit** from one of these classes:
  - Usually an AWT application inherits from java.awt.Frame class (in order to open/display a main window)
  - Usually a SWING application inherits from javax.swing.JFrame class (in order to open/display main window)

- In the constructor, additional GUI controls (JButtons, JTextFields, JRadioButtons etc) may be added to the class.
- After being instantiated as separate objects, they are connected to the JFrame by being added to the JFrame's content pane
- They may also be added into other objects (e.g. JPanels) before being added to the main content pane
- Each Panel type object usually has a layout manager associated, which can be used to control the arrangement of GUI controls or panel sub-objects

- Normally, if JFrame is being extended, one might also override the paintComponent(Graphics g) method, to define any specialized drawing that will go on within the application

2) What is a JFrame?

1. A top level component that represents a basic type of window for an application – it includes only the "frame" of the window (i.e. the bar/border of the window with its standard buttons for minimizing, maximising and exiting, the title bar, and can be grabbed to move or resized (if enabled).

3) Outline 2 different approaches to creating an application that employs JFrame(s). Are there any advantages to using one approach versus the other? Explain.

1. Instantiate a JFrame object and use it in an application:
   - No real advantage, other than you do not need to define a constructor

2. Extend the JFrame class:

- (adv): Can add fields directly to the JFrame object (to store components stored and added to the content pane) – this makes it easier to directly reference and modify/look at state of these gui components
- (adv): can override methods of JFrame class (e.g. paintComponent, etc) – allowing for a custom rendering behaviour that is automatically triggered from various events such as resizing etc..

4) What is a JPanel?
A component that acts as a container, generally used to hold other GUI components. JPanel is swing's version of a Panel (from AWT). It can be assigned a specific layout manager, usually used to group a set of GUI components together.

5) What is a UI Control?

An AWT/Swing component that has a particular visual representation allowing for direct interaction via mouse/keyboard. The object is automatically configured to change its visual representation and state as a result of that interaction (without the user having to do anything custom..). The processing of that state in the context of the application is usually not configured by default, and a user needs to add (register) a listener/handler to do this

6) In the context of Swing, How do you create a

i. CheckBox
ii. Radio Button Group
iii. Image Icon
iv. Image
v. Scrollable Panel including several other components
vi. Application with a Split Area (i.e. two areas left/right or top/bottom)

(see slides for these answers – you should know them in the context of Swing .. i.e. the J-version of components: JComponents and subclasses):

```
JCheckBox chinButton = new JCheckBox("Chin");          // L12, slide 18

ButtonGroup group = new ButtonGroup();                 // L12, slide 20
group.add(bird);                                       // assumes bird and cat are JRadioButtons
group.add(cat);

ImageIcon icon1 = new ImageIcon("images/3d_file.png"); // L12, slide 13
```

```
Image img = icon1.getImage();                    // if you create an image icon first..

JScrollPane scrollPane = new JScrollPane(jp);    // L13, slide 20 - jp is a JPanel with gui components

JSplitPane split = new JSplitPane(JSplitPane.HORIZONTAL_SPLIT,
                                  panelLeft, panelRight);        // L13, slide 22
```

7) In the context of Swing, how do you create:

   i. An object that is used as a "spacer" – used to fill an empty space of a given size and position

```
// see L16, slide 24
```

   ii. An application with all GUI controls laid out horizontally?

```
// L12, slide 32:   use a BoxLayout with BoxLayout.X_AXIS
```

   iii. An application with all GUI controls on a regular grid?

```
// L12, slide 38: use a GridLayout
```

   iv. A popup dialog window (message only)?

```
// see ButtonLabelDemo2.java in lect code for week 8
// (inside ButtonClickListener class).  Assume this is a JFrame class
// i.e. some parent window component for the dialog window (all dialogs must
// have a parent window)

JOptionPane.showMessageDialog(this, "some message");
```

   v. A popup dialog window (that accepts text input)?

```
String inputStr = JOptionPane.showInputDialog("some message");
```

   vi. A Shape (any type of geometry from java.awt.geom)?

```
// example:
Shape myLine = new Line2D.Double(x1,y1,x2,y2);
```

8) What is the purpose of a Layout Manager in Swing?

To automatically position and re-size JComponents (within the user space of a container) according to some specific, predefined policy

9) Explain how one would achieve absolute positioning of GUI components in Swing?

Configure the JFrame (or Container you are positioning GUI components within), to have a null layout, then use setBounds(..) methods on each component to define its position (x,y) and size (width, height) explicitly.

10) Outline how you would make a TextArea scrollable in Swing?

Instantiate a ScrollPane object using the TextArea object as an argument
Set the scroll policies for vertical/horizontal as needed

11) Outline how you would make a set of GUI controls all work within a common, scrollable area within part of a JFrame?

Put all of the controls into a new JPanel object, then add that JPanel into a ScrollPane object, then set the scroll policies as needed

12) How do we configure a JFrame to respond to:

i. Window resizing?
Use .setResizable(**true**);

ii. Window minimizing?
Already enabled on JFrame by default

iii. Exiting on clicking the 'x' button in the top of the Jframe?
Use .setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***);

13) How do you modify the state of the following UI Controls in Swing?

i. A TextField object ?
ii. A TextArea object?
iii. A JLabel object?
iv. An Icon within a JButton object?

// (i-iii) see L14
// (iv) create a new ImageIcon object and invoke .setImage(..) on JButton

14) Explain how one would access a JPanel object to draw graphics and geometries directly inside it?

// assume JPanel is called myPanel

Graphics g = myPanel.getGraphics();
Graphics2D g2d = (Graphics2D) g;

// use g to invoke drawXX/fillXX methods
// use g2d to invoke draw(Shape) & fill(Shape) methods

15) What are the benefits to extending a JComponent object (any component) versus just instantiating one and using it as a field or instance variable.

Can define custom rendering behaviour (and other behaviours)..
Override : paint(Graphics g) OR paintComponent(Graphics g) method..

As an instantiated object, we can only call the default versions of paint() or paintComponent()