

Assignment no. 3

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

Q. 2) What is JButton? Write down simple Java program to illustrate more about JButton.

-
- The JButton class is used to create a labelled Button and that has platform independent implementation.
 - The application results in some action when the button is pushed.
 - Syntax:-

public class JButton extends AbstractButton
implements Accessible

• Program:-

```
import javax.swing.*;  
public class ButtonEx{  
    public static void main (String args[]){  
        JFrame f = new JFrame ("Button");  
        JButton b = new JButton ("submit");  
        b.setBounds (50,50,100,50);  
        f.add (b);  
        f.setLayout (null);  
        f.setSize (300,300);  
        f.setVisible (true);  
    }  
}
```

Output

• Output:-

submit

Q.3)

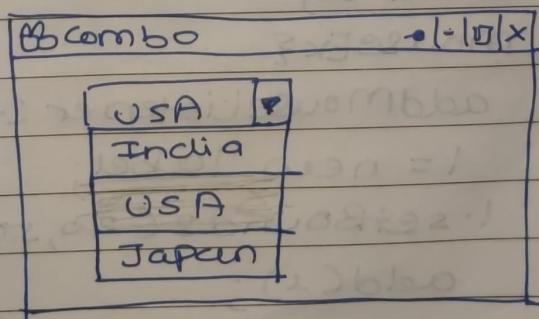
What is JComboBox? write down a simple Java program to illustrate more about JComboBox.

- • The JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false).
- Clicking on a checkbox
- The object of choice class is used to show popup menu of choices. Choice selected by user is shown on the top of the menu.

Program:-

```
import javax.swing.*;
public class JComboBoxEx {
    public static void main (String args []){
        JFrame f = new JFrame ("Combo");
        String country [] = {"India", "USA", "Japan"};
        JComboBox cb = new JComboBox (country);
        cb.setBounds (50, 50, 90, 20);
        f.add (cb);
        f.setLayout (null);
        f.setSize (300, 400);
        f.setVisible (true);
    }
}
```

Output :-



Q.4) What is Mouse Event Handling?

Methods of mouse Listener Interface?

Write down a simple program.



- The Java mouseListener is notified whenever you change the state of mouse.
- It is notified against Mouse event.
- Methods of Mouse Listener Interface-
 - a) public abstract void mouseClicked (MouseEvent e);
 - b) public abstract void mouseEntered (MouseEvent e);
 - c) public abstract void mouseExited (MouseEvent e);
 - d) public abstract void mousePressed (MouseEvent e);
 - e) public abstract void mouseReleased (MouseEvent e);

• Program:-

```

import java.awt.*;
import java.awt.event.*;
public class mouseEx extends Frame
  implements MouseListener {
  Label l;
  mouseEx {
    addMouseListener(this);
    l = new Label();
    l.setBounds(20, 50, 100, 20);
    add(l);
    setSize(300, 300);
    setLayout(null);
  }
}
  
```

```

        setVisible(true);
    }

    public void mouseClicked(MouseEvent e) {
        l.setText("Mouse Clicked");
    }

    public void mouseEntered (MouseEvent e) {
        l.setText ("Mouse Entered");
    }

    public void mouseExited (MouseEvent e) {
        l.setText ("Mouse Exited");
    }

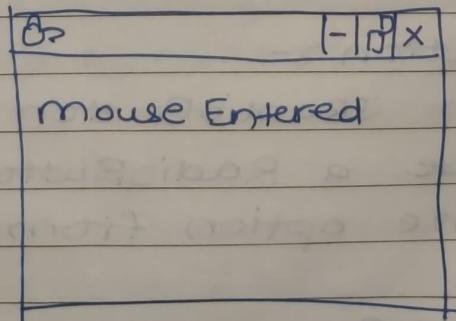
    public void mousePressed (MouseEvent e) {
        l.setText ("Mouse Pressed");
    }

    public void mouseReleased (MouseEvent e) {
        l.setText ("Mouse Released");
    }

    public static void main (String args []) {
        new mouseEx();
    }
}

```

Output:-



Q.1) List down GUI components used in swing?

⇒ • JButton :-

JButton object class is used to create a label of a button. It provides independent implementation and it inherits from AbstractButton class.

• JLabel :-

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text.

• JFrame :-

JFrame class is type of container which inherits the java.awt.Frame class. JFrame works like the main window.

• JCheckBox :-

JCheckBox class is used to create a checkbox. It is used to turn an option on (true) or off (false).

• JRadioButton :-

JRadioButton class is used to create a RadioButtton. It is used to choose one option from multiple options.

• JComboBox :-

The object of choice class is

used to show popup menu of choices.

- JList :-

The object of JList class represents a list of item.

Q.5) What is Multiple Selection List? Write down a simple program to illustrate more about multiple selection list.

- • Multiple selection list is same as JList.
- Instead, it allows us to select many items from a JList.
- Program :-

```

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import com.online
public class JListCopy extends JFrame{
    private JList jlist;
    private JList forJListCopy, jListForCopy;
    private static final String[] listItems =

```

```
{ "Blue", "Black", "Cyan", "Green", "Gray",
  "Red", "White" };
```

```
private static final Color[] colors = { Color.Blue,
  Color.Black, Color.Cyan, Color.Green, Color.Gray,
  Color.Red, Color.White };
```

```
public JList copy() {
```

```
super ("JList Demo");
```

```
setLayout (new FlowLayout ());
```

```
jlist = new JList (listItems);
```

```
jlist.setFixedCellHeight (15);
```

```
jlist.setFixedCellWidth (100);
```

```
jlist.setSelectionMode (ListSelectionModel.
```

```
SINGLE_SELECTION);
```

```
jlist.setVisibleRowCount (4);
```

```
add (new JScrollPane (jlist));
```

```
copyButton = new JButton ("Copy >");
```

```
copyButton.addActionListener (new ActionListener () {
```

```
{
```

```
@Override
```

```
public void actionPerformed (ActionEvent e) {
```

```
  jlistForCopy.setListData (jlist.getSelectedValues ());
}
```

```
} );
```

```
add (copyButton);
```

```
jlistForCopy = new JList ();
```

```
jlistForCopy.setFixedHeight (18);
```

```
jlistForCopy.setFixedWidth (100);
```

```
jlist.setVisibleRowCount (4);
```

```
jlist.setSelectionMode (ListSelectionModel.
```

```
MULTIPLE_INTERVAL_SELECTION);
```

```
add (new JScrollPane (jlistForCopy));
```

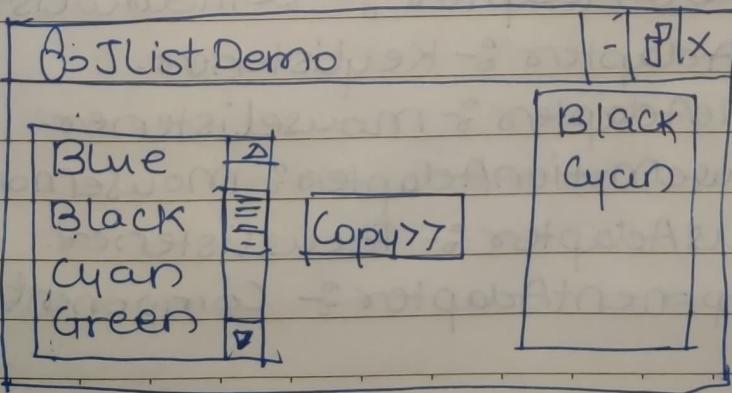
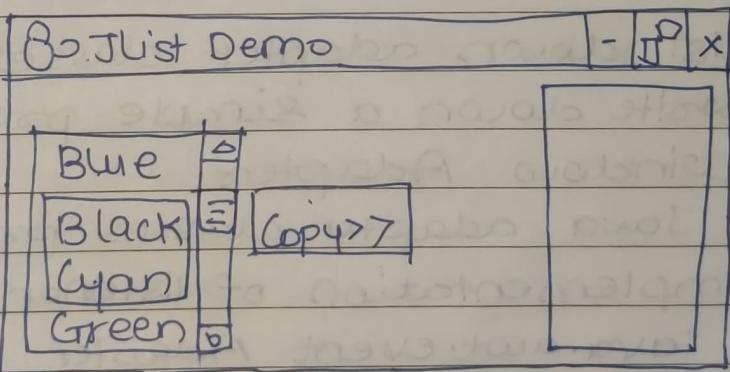
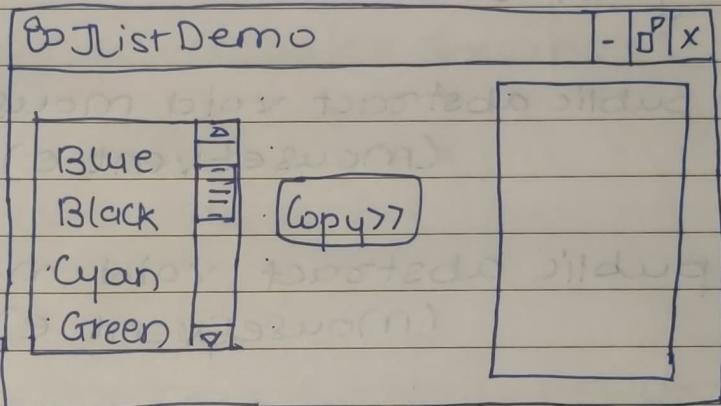
```
}
```

```

public static void main (String args[])
{
    jlistDemo
    jlistCopy jListDemo = new JListCopy ();
    jListDemo.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    jListDemo.setSize (350, 150);
    jListDemo.setVisible (true);
}
}

```

• Output:-



Q.6) What are methods used in Mouse Motion Listener Interface?

- • The Java MouseMotionListener is notified whenever you move or drag mouse.
- It is notified against MouseEvent.
- The MouseMotionListener interface int is found in java.awt.event package.
- It has two methods.
- The signature of 2 methods found in ~~Mouse~~ MouseMotionListener interface are given below:

1. public abstract void mouseDragged (MouseEvent e);

2. public abstract void mouseMoved (MouseEvent e);

Q.7) List down adapter classes in swing & write down a simple program for Window Adapter.

- • Java adapter classes provides the default implementation of listener interfaces.
- java.awt.event Adapter classes:-
 a) WindowAdapter :- WindowListener.
 b) KeyAdapter :- KeyListener.
 c) MouseAdapter :- MouseListener.
 d) MouseMotionAdapter :- MouseMotionListener.
 e) FocusAdapter :- FocusListener.
 f) ComponentAdapter :- ComponentListener.

- g) ContainerAdapters :- ContainerListeners.
- h) HierarchyAdapters & HierarchyListeners.

- java.awt.dnd Adapter Classes :-

- a) DragSourceAdapter :- DragSourceListeners.
- b) DragTargetAdapter :- DragTargetListeners.

- javax.swing.event Adapter classes :-

- a) MouseInputAdapter :- MouseInputListeners.
- b) InternalFrameAdapter :- InternalFrameListeners.

- Java WindowAdapter Example :-

```

import java.awt.*;
import java.awt.event.*;
public class AdapterEx {
    Frame f;
    AdapterEx () {
        f=new Frame ("Window Adapter");
        f.addActionListener (new WindowAdapter () {
            public void windowClosing (WindowEvent e) {
                f.dispose ();
            }
        });
        f.setSize (400,400);
        f.setLayout (null);
        f.setVisible (true);
    }
}

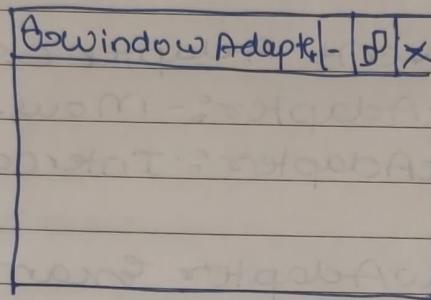
```

```

public static void main(String args[]){
    new AdapterEx();
}
}

```

• Output :-



Q. 8) What is Meant by Key Event Handling?
Write down a simple program for counting total words & characters entered in Java TextField?

- • The Java KeyListener is notified whenever you change the state of key.
 - It is notified against KeyEvent.
 - The KeyListener interface is found in java.awt.event package.
 - It has 3 methods:-
- a) public abstract void keyPressed(KeyEvent e);
 - b) public abstract void keyTyped(KeyEvent e);
 - c) public abstract void keyReleased(KeyEvent e);

• Program :-

```

import java.awt.*;
import java.awt.event.*;
public class KeyListenerEx extends Frame
    implements KeyListener {

```

```

import Label l;
TextField tf;
KeylistenerEx() {
    l = new Label();
    l.setBounds(20, 50, 200, 20);
    tf = new TextField();
    tf.setBounds(20, 80, 300, 300);
    l.add(tf);
    tf.addKeyListeners(this);
    add(l);
    add(tf);
    setSize(400, 400);
    setLayout(null);
    setVisible(true);
}

```

```

public void KeyPressed(KeyEvent e) {}
public void KeyReleased(KeyEvent e) {} {
    String text = tf.getText();
    String words[] = text.split("\\s");
    l.setText("Words:" + words.length + "Characters:" +
        text.length());
}

```

```

public void KeyTyped(KeyEvent e) {}
public static void main(String args[]) {
    new KeyListenerEx();
}

```

Output :-

8	1-D X
Words: 2 Character: 13	
manish Jadhav	

Q.9) What are different layout managers used in swing? write down a simple program using any layout manager class.

- • The LayoutManagers are used to arrange components in particular manner.
- There are following ~~classes that represents~~ types of Layout managers:-
- a) BorderLayout
 - b) FlowLayout
 - c) CardLayout
 - d) GridLayout
 - e) GridBagConstraints
 - f) BoxLayout
 - g) GroupLayout
 - h) ScrollPaneLayout
 - i) SpringLayout

• Program on GridLayout :-

```

import java.awt.*;
import javax.swing.*;

public class MyGridLayout {
    JFrame f;
    MyGridLayout() {
        f = new JFrame();
        JButton b1 = new JButton("Volvo");
        JButton b2 = new JButton("Festina");
        JButton b3 = new JButton("Audi");
        JButton b4 = new JButton("BMW");
        f.add(b1);
        f.add(b2);
        f.add(b3);
        f.add(b4);
        f.setLayout(new GridLayout(2, 2));
    }
}

```

```
f.setSize(300, 300);
f.setVisible(true);
}
```

```
public static void main(String args[]){
    new GridLayout();
    }
}
```

• Output :-

Bp		- D X
Velar	Fortuner	
Audi	BMW	

Q.10) What is meant by JTextArea? Write down a simple program to Count words and characters.

- The object of TextArea class is a multi line region that displays text.
- It allows the editing of multiple line text.
- It inherits JTextComponent class.
- Programs:-

```
(Ways print2) import javax.swing.*;
import java.awt.event.*;
public class TextAreaEx implements ActionListener {

```

```
JLabel l1, l2;
JTextArea a;
```

```

JButton b;
TextArea TextAreaEx () {
    JFrame f = new JFrame ();
    l1 = new JLabel ();
    l1.setBounds (50, 25, 100, 30);
    l2 = new JLabel ();
    l2.setBounds (160, 25, 100, 30);
    a = new JTextArea ();
    a.setBounds (20, 75, 250, 200);
    b = new JButton ("Count");
    b.setBounds (100, 300, 120, 30);
    b.addActionListener (this);
    f.add (l1); f.add (l2); f.add (a);
    f.add (b);
    f.setLayout (null);
    f.setSize (450, 450);
    f.setVisible (true);
}

public void actionPerformed (ActionEvent e) {
    String text = a.getText ();
    String words [] = text.split ("\n");
    l1.setText ("Words: " + words.length);
    l2.setText ("Characters: " + text.length ());
}

public static void main (String args []) {
    new TextAreaEx ();
}

```

• Output:-

85	1-10 X
Words: 2	Character: 13
Manish Jadhav	