

## Advance Java Programming Assignment 1

**Q1. What components will be needed to get following output? 2 Marks**



- a) Label, TabbedPane, CheckBox
- b) TabbedPane, List, Applet
- c) Panel, TabbedPane, List
- d) Applet, TabbedPane, Panel

**Q2) Select the missing statement in given code 2 Marks**

```
// Demonstrate the mouse event handlers.
import java.awt.*;
import java.applet.*;
/*
<applet code="mouse" width=300 height=100>
</applet>
*/
public class mouse extends Applet
implements MouseListener, MouseMotionListener
{
String msg = "";
int mouseX = 0, mouseY = 0; // coordinates of mouse
public void init() {
}
// Handle mouse clicked.
public void mouseClicked(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}
// Handle mouse entered.
public void mouseEntered(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
// Handle mouse exited.
public void mouseExited(MouseEvent me)
{
mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
// Handle button pressed.
```

```

public void mousePressed(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
// Handle button released.
public void mouseReleased(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
// Handle mouse dragged.
public void mouseDragged(MouseEvent me)
{
mouseX = me.getX();
mouseY = me.getY();
msg = "*";
showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
repaint();
}
// Handle mouse moved.
public void mouseMoved(MouseEvent me)
{
showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}
// Display msg in applet window at current X,Y location.
public void paint(Graphics g)
{
g.drawString(msg, mouseX, mouseY);
}
}

```

- a)addMouseListener(this);
- b)addMouseListener(this);
- addMouseMotionListener(this);
- import java.awt.event.\*;
- c) addMouseListener();
- d) all of above

**Q3) Draw the Output**

```

Import java.awt.event.*;
import java.awt.*;
import java.applet.*;
public class checkbackg extends Applet implements ItemListener
{
Checkbox m1,m2,m3;
public void init()
{
m1=new Checkbox("A");
m2=new Checkbox("B");
m3=new Checkbox("C");
}
}

```

```

add(m1);
add(m2);
add(m3);
m1.addItemListener(this);
m2.addItemListener(this);
}
public void itemStateChanged(ItemEvent ie)
{
if(ie.getSource()==m1)
setBackground(Color.red);
if(ie.getSource()==m2)
setBackground(Color.green);
}
}
/*<applet code=checkbackg.class height=150 width=150>
</applet>*/

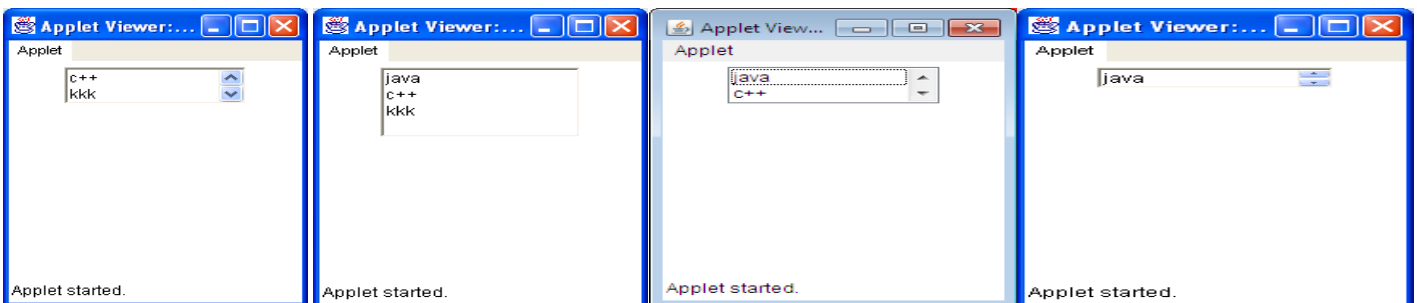
```

**Q4) select the proper output for following code 2 Marks**

```

import java.awt.*;
import java.applet.*;
public class list2 extends Applet
{
public void init()
{
List l= new List(2,true);
l.add("java");
l.add("c++");
l.add("kkk");
add(l);
}
}
/*<applet code=list2.class height=200 width=200>
</applet>*/

```



**Q5) Debug the following program 2 Marks**

```

import java.awt.*;
import javax.swing.*;
/*
<applet code="JTableDemo" width=400 height=200>
</applet>
*/
public class JTableDemo extends JApplet
{
public void init() {
Container contentPane = getContentPane();

```

```

contentPane.setLayout(new BorderLayout());
final String[] colHeads = { "emp_Name", "emp_id", "emp_salary" };
final Object[][] data = {
{ "Ramesh", "111", "50000" },
{ "Sagar", "222", "52000" },
{ "Virag", "333", "40000" },
{ "Amit", "444", "62000" },
{ "Anil", "555", "60000" },
};
JTable table = new JTable(data);
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane jsp = new JScrollPane(table, v, h);
contentPane.add(jsp, BorderLayout.CENTER);
}
}

```

- a. Error in statement in which JTable is created
- b. Error in statement in which JScrollPane is created
- c. Error in statement in which applet tag is declared
- d. None of the above

**Q6) Draw the output**

// Demonstrate the key event handlers.

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="SimpleKey" width=300 height=100>
</applet>
*/
public class SimpleKey extends Applet
implements KeyListener {

String msg = "";
int X = 10, Y = 20; // output coordinates

public void init() {
addKeyListener(this);
requestFocus(); // request input focus
}

public void keyPressed(KeyEvent ke) {
showStatus("Key Down");
}

public void keyReleased(KeyEvent ke) {
showStatus("Key Up");
}

public void keyTyped(KeyEvent ke) {
msg += ke.getKeyChar();
repaint();
}

// Display keystrokes.

```

```

public void paint(Graphics g) {
    g.drawString(msg, X, Y);
}
}

```

**Q7) Explain the program**

```

// Anonymous inner class demo.
import java.applet.*;
import java.awt.event.*;
/*
<applet code="AnonymousInnerClassDemo" width=200 height=100>
</applet>
*/
public class AnonymousInnerClassDemo extends Applet {
    public void init() {
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent me) {
                showStatus("Mouse Pressed");
            } });
    }
}

```

**Q8) A label is a simple control which is used to display \_\_\_\_\_ on the window:**

- a. Text(non-editable)
- b. Text(editable)
- c. Both a & b
- d. None of these

**Q9) What is the highest-level event class of the event-delegation model?**

Ans

**Q10). Which method is used to translate a mouse click on a specific point of the tree to a tree path?**

- a) translatePoint( )
- b) getLocation( )
- c) getPathForLocation( )
- d) getPath( )

**Q11) Draw the Output**

```

import java.awt.*;
import java.applet.*;
/*
<applet code="GridLayoutDemo" width=300 height=200>
</applet>
*/
public class GridLayoutDemo extends Applet {
    static final int n = 4;
    public void init() {
        setLayout(new GridLayout(n, n));

        setFont(new Font("SansSerif", Font.BOLD, 24));

        for(int i = 0; i < n; i++) {
            for(int j = 0; j < n; j++) {
                int k = i * n + j;
                if(k > 0)

```

```
        add(new Button("" + k));  
    }  
}  
}
```