

Unit : 02

Swings

1) Swing is a set of classes that provides more _____ and _____ components than are possible with the AWT

a) **Powerful , Flexible**

b) Great, Awesome

c) Better , Powerful

d) Stable, customizable

2) Swing components are not implemented by _____ code.

a) **platform-Specific**

b) Platform- independant

c) Both

d) none

3) Swing Components are _____ Weight

a) **Light**

b) Moderate

c) Heavy

d) d)All of the Above

4) Super Class For Swing Buttons

a) **AbstractButton**

b) Button

c) Jbutton

d) none

5) Which class encapsulates a mutually exclusive set of buttons.

a) MutualButtonGroup

b) **ButtonGroup**

c) JButtonGroup

d) ButtonsGroup

6) Which class encapsulates an icon.

a) Image

b) Icon

c) ImageIcon

d) **ImageIcon**

7) Which class represents the Swing version of Applet.

a) SwingApplet

b) SApplet

c) **JApplet**

d) JSApplet

8) Which is Swing push button class.

a) JPushButton

b) PushButton

c) Jbutton

d) **None**

9) Which is the Swing check box class.

a) JCheckbox

b) JSCheckbox

c) **JCheckBox**

d) None

10) JApplet is ___ with functionality when compared with Applet

a) **Rich**

b) poor

- c) Same
- d) different

11) Which class encapsulates a Swing combo box

- a) JComboBox
- b) jCombox
- c) **JComboBox**
- d) none

12) Which package needs to be imported for using Swing Classes

- a) java.swing
- b) java.applet.swing
- c) **javax.swing**
- d) java.lang.swing

13) The Swing version of a label.

- a) SwingLabel
- b) Label
- c) JLabel
- d) **none**

14) The method used to add components to a Container

- a) **add()**
- b) Insert()
- c) addComponent
- d) InsertComponent

15) The Swing version of a radio button.

- a) **JRadioButton**
- b) JRadiobutton
- c) JradioButton
- d) none

16) Which class Encapsulates a scrollable window in Swing

- a) JScrollableWindow
- b) JScrollablewindow
- c) JscrollableWindow
- d) **JScrollPane**

17) The Components in JApplet are added to

- a) Current JApplet Class
- b) **ContentPane**
- c) both
- d) none

18) Which class encapsulates a tabbed window.

- a) JTabbedWindow
- b) jTabbedWindow
- c) JtabbedWindow
- d) **none**

19) Class which Encapsulates a table-based control

- a) JControlTable
- b) **JTable**
- c) JcontrolTable
- d) JControltable

20) The Swing classes of a text field.

- a) **JTextField**
- b) JTextField
- c) Both
- d) None

21) For using JTree one has to import javax.swing.tree separately as only javax.swing.* does not work

- a) **True**
- b) Partially True
- c) False
- d) Partially False

22) Constructors for using Icon and Label In Swing

- a) ImageIcon(String filename)
- b) ImageIcon(URL url)
- c) **Both**
- d) None

23) int getIconHeight() Returns the height of the icon as

- a) **centimetres**
- b) millimetres
- c) pixels
- d) none

24) The Constructor used to create a JTextField with predefined Text

- a) **JTextField(String s, int cols)**
- b) void setText(String)
- c) both

d) none

25) JRadioButton class, which is a concrete implementation of _____

a) Container

b) **AbstractButton**

c) Both

d) none

26) Items are added to the JComboBox using _____ method

a) add()

b) **addItem()**

c) addOption()

d) none

27) For a JComboBox Constructor which can be the arguments passed

a) Array

b) **Vector**

c) Both

d) None

28) Which are the methods used for JLabel

Icon getIcon()

String getText()

void setIcon(Icon i)

void setText(String s)

a) **All**

b) Only Icon getIcon() and String getText()

c) Both

d) None

29) Constructors for JTextField

a)

JTextField()

JTextField(int cols)

JTextField(String s, int cols)

JTextField(String s)

b)Only JTextField() and JTextField(int cols)

c) both

d) none

30) Constructors for JCheckBox

a)

JCheckBox(Icon i)

JCheckBox(Icon i, boolean state)

JCheckBox(String s)

JCheckBox(String s, boolean state)

JCheckBox(String s, Icon i)

JCheckBox(String s, Icon i, boolean state)

b)Only JCheckBox(Icon i) and JCheckBox(Icon i, boolean state)

c)both

d)none

31) For JLabel alignment Constants are

a) **LEFT, RIGHT, CENTER, LEADING, TRAILING**

b) Only LEFT, and RIGHT,

c) Both

d) None

32) Constructors for JLabel

a) **JLabel(Icon i),Label(String s),JLabel(String s, Icon i, int align)**

b) Only JLabel(Icon i) and Label(String s)

c) Both

d) done

33) Components such as buttons have _____ capabilities in Swing.

a) **More**

b) Less

c) Equal

d) No

34) An Image can be changed as the state of the button changes. True or False?

a) **True**

b) False

35) What is an AbstractButton class?

a) **Abstract Superclass for Swing**

b) Abstract subclass for Swing

c) Abstract Superclass for AWT

d) Abstract Subclass for AWT

36) Which class encapsulates a mutually exclusive set of buttons?

a) GroupButton

- b) **ButtonGroup**
- c) GroupingButton
- d) ButtonGrouping

37) Which class is used to manage buttons in Swing?

- a) SwingButton
- b) Button
- c) **JButton**
- d) JSwingButton

38) AbstractButton class extends which class?

- a) **JComponent**
- b) AbstractComponent
- c) ButtonComponent
- d) Component

39) void setDisabledIcon(Icon di) - in this method, what is the work of “Icon di”?

- a) This is the component that will be disabled
- b) It is the icon that will be disabled
- c) **It is an icon that is displayed when a component is disabled**
- d) It deletes the icon displayed on the screen

40) Which Listener is used with subclasses of AbstractButton to generate events?

- a) **ActionListener**
- b) ItemListener
- c) AdjustmentListener
- d) KeyListener

41) AbstractButton is a superclass for which components?

- a) Combo boxes, check boxes, and lists
- b) push buttons, check boxes, and radio buttons.**
- c) Text fields, Text areas
- d) None of the above

42) Which of the following is the correct constructor for JButton?

- a) JButton(String s, Icon i)**
- b) JButton(Icon i, String s)
- c) JButton(String s, Icon i, JButton.LEFT)
- d) None of the above

43) Which are the constructors of JButton class?

- a) JButton(Icon i) , JButton(String s) , JButton(String s, Icon i)**
- b) Only JButton(String s) ,JButton(String s, Icon i)
- c) Only JButton(String s, Icon i)
- d) Jbutton(icon i), Jbutton(string s), Jbutton(string s, icon i)

44) Which class is used to add Radio Buttons in swing?

- a) RadioButtonSwing
- b) RadioButton
- c) SwingRadioButton
- d) JRadioButton**

45) What is the immediate superclass of JRadioButton class?

- a) JToggleButton**
- b) JButton
- c) JAbstractButton
- d) JButtonMain

46) Which is the correct syntax for constructor of JRadioButton?

- a) JRadioButton(Icon i, String s, boolean state)
- b) JRadioButton(String s, Icon i, boolean state)**
- c) JRadioButton(Icon i,boolean state, String s,)
- d) d)JRadioButton(boolean state, String s, Icon i)

47) How many options can be selected at a time in JRadioButton?

- a) Only one**
- b) Multiple
- c) It is defined in the program
- d) Only a maximum of six

48) Which methods is used to get the text associated with a Radio Button?

- a) getText()
- b) getActionCommand()**
- c) getradioButtonText()
- d) getAction()

49) How many options can be selected at a time in a JCheckBox?

- a) Only one
- b) It is defined in the program
- c) As many as you want**
- d) JCheckBox isn't a Swing component

50) What will be the correct statement if you want a checkbox to be checked by default?

- a) JCheckBox jc= new JCheckBox(String s, boolean checked)
- b) JCheckBox jc= new JCheckBox(String s, boolean false)
- c) JCheckBox jc= new JCheckBox(String s, boolean unchecked)
- d) JCheckBox jc= new JCheckBox(String s, boolean true)**

51) Item Event for JCheckBox is handled by which method?

- a) ItemStateChanged()
- b) itemChanged()
- c) itemStateChange()
- d) **None of the above**

52) Which method is used to get the text of a check box?

- a) GetText()
- b) getCheckboxText()
- c) **getText()**
- d) GetCheckboxText()

53) Which method can be used to change the state of the JCheckBox?

- a) void setSelector(boolean state)
- b) **void setSelected(boolean state)**
- c) void Setsselected(boolean state)
- d) void setSelectedItem(boolean state)

54) JTabbed Pane manages a set of _____ by linking them with _____

- a) links,GUI
- b) **components,tabs**
- c) data,constructors
- d) controls,GUI

55) JTabbed pane uses

- a) **SingleSelectionModel**
- b) MutipleSelectionModel

c) MVC Architecture

d) None of these

56) Form of adding Tab

a) **void addTab(String name, Component comp)**

b) void addtab(String name, Component comp)

c) void addTab()

d) void AddTab(Stringname, Componentcomp)

57) Tabs are added by calling

a) getTab()

b) setTab()

c) **addTab()**

d) all of these

58) Here ,the component is added to a tab is

a) **JPanel**

b) JFrame

c) JTextField

d) All of these

59) Constructor(s) of JScrollPane

a) **JScrollPane(Component comp)**

b) ScrollPane(Componentcomp)

c) JScrollPane(Component string)

d) All of these

60) A tree presents _____ view of data

a) Structural

b) **Hierarchical**

c) Multiple

d) Single

61) Constructors of jtree are

a) JTree(Object obj[])

JTree(Vec)

JTree(TreeNode tn)

b) JTree(Object obj[])

JTree(Vector<?>v)

JTree(Treenode tn)

c) JTree(Object obj[])

JTree(Vector<?> v)

JTree(TreeNode tn)

d) None of these

62) The tree event classes and listener interfaces are packaged in

a) JAva.Swing.*;

b) javaSwing.;

c) java;

d) javax.Swing.event

63) You can obtain the path to the selected object by calling _____ shown here, on the event object _____

a) getPath(), path p

b) getPath(), TreePath getPath()

c) setpath, TreePath

d) None of these

64) The _____ interface extends TreeNode

- a) **MutableTreeNode**
- b) DefaultMutableTreeNode
- c) Both a & b
- d) None of these

65) _____ is a component that displays rows and columns of data.

- a) JPanel
- b) JFrame
- c) **JTable**
- d) None of these

66) Constructor used for JTable is

- a) **JTable(Objectdata[][], ObjectcolHeads[])**
- b) JTable(Objectdata[][], ObjectcolHeads[])
- c) JTable()
- d) All of these

67) MVC stands for

- a) Model-View-Control
- b) Menu-View-Control
- c) Menu-Visual-Control
- d) **Model-View-Controller**

68) In MVC terminology, the model corresponds to the

- a) Components and links
- b) **state information associated with the component.**

- c) Data
- d) All of these

69) The _____ determines how the component reacts to the user

- a) View
- b) **Controller**
- c) User
- d) Model

70) In which package Swing components are defined?

- a) javax.applet.swing
- b) **javax.swing**
- c) java.javax.swing
- d) javax.java.swing

71) The super class of all swing buttons is –

- a) Button
- b) ButtonGroup
- c) JButton
- d) **AbstractButton**

72) Which of the following alignment is not possible for JLabel?

- a) **TOP**
- b) LEFT
- c) CENTER
- d) LEADING

73) Alignment constants of JLabel are the part of –

a) **SwingConstants interface**

b) SwingConstants class

c) Swing class

d) Graphics class

74) How will you set icon for the JLabel?

a) Using Icon class directly

b) **Using setIcon() method**

c) Using makeIcon() method

d) It is not possible to set icon for JLabel

75) Swing's text field is encapsulated by –

a) Component class

b) JComponent class

c) Container class

d) **JTextComponent class**

76) How to give number of columns for JTextField?

a) Use setColumns() method

b) **Use the value directly in the constructor**

c) Using applyColumn() method.

d) We have to use JTextArea class

77) What is the return type of getText() method of JButton class?

a) void

b) **String**

c) Character array

d) There is no such method

78) How will you assign the string and icon both to the JButton?

- a) It is not possible
- b) Use the setTextIcon() method
- c) Use the setIconText() method
- d) **Initialize them directly in the constructor**

79) Which event is generated when a JButton is pushed?

- a) ItemEvent
- b) TextEvent
- c) PushEvent
- d) **ActionEvent**

80) Immediate super class of JCheckBox is –

- a) JComponent
- b) JApplet
- c) JCheckBoxGroup
- d) **JToggleButton**

81) The constructor JCheckBox(true, “YES”) suggests that –

- a) **Checkbox is selected and displays the string “YES” on it.**
- b) Checkbox is deselected and displays the string “YES” on it.
- c) Checkbox is selected and it shows the tick always on it.
- d) There is no such constructor.

82) When JCheckBox is clicked the _____ event is generated.

- a) **ItemEvent**
- b) ActionEvent
- c) TextEvent
- d) MouseEvent

83) How can we create Radio buttons in Swing?

- a) Using ButtonGroup class
- b) Using JCheckboxGroup class
- c) **Using JRadioButton class**
- d) Using JButton class

84) How to make the group of Radio buttons?

- a) **Using ButtonGroup class**

- b) Using JButtonGroup class
- c) Using JRadioButton
- d) Using AbstractButton class

85) How to contents of whole vector into the JComboBox?

a) **Use the constructor of JComboBox**

- b) Use method addItem()
- c) Use method addVector()
- d) Use method addValues()

86) Which method is used to add the tabs in the tabbed pane?

- a) add()
- b) addItem()
- c) addPane()
- d) **addTab()**

87) The scroll bar constants for scroll pane are defined in –

- a) Scrollbar class
- b) ScrollPane class
- c) **ScrollPaneConstants class**
- d) Component class

88) Which of the following constants shows scroll bars always?

- a) **HORIZONTAL_SCROLLBAR_ALWAYS**
- b) HORIZONTAL_SCROLLBAR_AS_NEEDED
- c) HORIZONTAL_SCROLLBARS
- d) HORIZONTAL_ALWAYS

89) JScrollPane is an immediate sub-class of –

- a) JContainer
- b) JApplet
- c) JComponent
- d) ScrollPane

90) Is it possible to add array of objects to trees? How?

- a) Not possible
- b) Yes, using its one of the forms of constructor**
- c) Yes, using the add() method
- d) Yes, using the addItem() method

91) When tree is expanded, which event is generated?

- a) ExpansionEvent
- b) TreeExpansionEvent**
- c) ItemEvent
- d) ActionEvent

92) When the user selects or deselects a node within the tree which event is generated.

- a) ExpansionEvent
- b) TreeExpansionEvent
- c) ItemEvent
- d) TreeSelectionEvent**

93) When the data or structure of the tree changes which event generates:

- a) ExpansionEvent
- b) TreeExpansionEvent
- c) TreeModelEvent**

d) TreeSelectionEvent

94) We can obtain the path to the selected object in tree by calling _____ method.

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

95) The TreeNode is –

- a) A class
- b) **An interface**
- c) A variable
- d) Nothing

96) The TreeExpansionEvent class is defined in –

- a) java.awt package
- b) javax.swing package
- c) java.awt.event package
- d) **javax.swing.event package**

97) TreeExpansionListener interface provides following method –

- a) getExpanded() **b) treeExpanded()**
- c) expanded() d) None of the above

98) How to create for Vector elements?

- a) **Pass vector as parameter for JTree**
- b) Use method addElements() for JTree class.

- c) Use method addVector() method of JComponent class
- d) It is not possible

99) Which of these are the subclasses of Toggle Button class?

- a) JRadioButton
- b) JCheckBox
- c) **Both a & b**
- d) None of these

100) Find the Errors

```
import java.awt.*;
import java.swing.*;

/*<applet code="JLabelDemo" width=250 height=150>
</applet>

*/

public class JLabelDemo extends JApplet {
public void init() {
Container contentPane = getContentPane();
ImageIcon ii = new ImageIcon("france.gif");
JLabel jl = new JLabel("France", ii, JLabel.CENTER);
contentPane.add(jl);
}
}
```

- a) Its Correct
- b) **Wrong package**
- c) Wrong constructor for creating Icon
- d) Both b and c

101) Error time :->

```
import java.awt.*;
import javax.swing.*;
/*
<applet code="JTextFieldDemo" width=300 height=50>
</applet>
*/
public class JTextFieldDemo extends JApplet {
    JTextField jtf;
    public void init() {
        Container contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());
        jtf = new JTextField(15,"Enter Text Here");
        contentPane.add(jtf);
    }
}
```

- a) **Program Code is correct**
- b) Wrong package imported
- c) Wrong constructor for TextFiled
- d) Wrong class extended

102)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
/*
<applet code="JComboBoxDemo" width=300 height=100>
</applet>
*/
```

```
public class JComboBoxDemo extends JApplet
implements ActionListener {
    JLabel jl;
    ImageIcon france, germany, italy, japan;
    public void init() {
        JComboBox jc = new JComboBox();
        jc.addItem("France");
        jc.addItem("Germany");
        jc.addItem("Italy");
        jc.addItem("Japan");
        jc.addItemListener(this);
        contentPane.add(jc);
        jl = new JLabel(new ImageIcon("france.gif"));
        contentPane.add(jl);
    }
    public void itemStateChanged(ItemEvent ie) {
        String s = (String)ie.getItem();
        jl.setIcon(new ImageIcon(s + ".gif"));
    }
}
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout);
JComboBox jc = new JComboBox();
jc.addItem("France");
jc.addItem("Germany");
jc.addItem("Italy");
jc.addItem("Japan");
jc.addItemListener(this);
contentPane.add(jc);
```



```

jl = new JLabel(new ImageIcon("france.gif"));
contentPane.add(jl);
}
public void itemStateChanged(ItemEvent ie) {
String s = (String)ie.getItem();
jl.setIcon(new ImageIcon(s + ".gif"));
}
}

```

- a)Wrong Listener Implemented** b)Program code is correct
c)Constructor for setting Layout is wrong d)Both a and c

103) Fill in the empty spaces in the following program so that the program doesn't have any errors:

```

import java.awt.*;
import java.awt.____.*;
import____.swing.*;
/*
<applet code="JButtonDemo" width=250 height=300>
</applet>
*/
public class JButtonDemo extends JApplet
implements ActionListener {
JTextField jtf;
public void init() {
// Get content pane
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout());
// Add buttons to content pane

```

```

ImageIcon france = new ImageIcon("france.gif");
JButton jb = new _____(france);
jb.setActionCommand("France");
jb.addActionListener(this);
contentPane.add(jb);

ImageIcon germany = new ImageIcon("germany.gif");
jb = new JButton(germany);
jb.setActionCommand("Germany");
jb.addActionListener(this);
contentPane.add(jb);

ImageIcon italy = new ImageIcon("italy.gif");
jb = new JButton(italy);
jb.setActionCommand("Italy");Chapter 26:
jb.addActionListener(this);
contentPane.add(jb);

ImageIcon japan = new ImageIcon("japan.gif");
jb = new JButton(japan);
jb.setActionCommand("Japan");
jb.addActionListener(this);
contentPane.add(jb);

// Add text field to content pane
jtf = new JTextField(15);
_____.add(jtf);
}

public void actionPerformed(ActionEvent ae) {
jtf.setText(ae.getActionCommand());
}
}

```

- a) **event, javax, JButton, contentPane**
- b) event, Java, Button, cpp
- c) awt, Javax, JComboBox, contentPane.
- d) Swing, event, awt, JTextField

104) What will be the packages required for this program to work?

```
/*  
<applet code="JRadioButtonDemo" width=300 height=50>  
</applet>  
*/  
  
public class JRadioButtonDemo extends JApplet  
implements ActionListener {  
    JTextField tf;  
    public void init() {  
        // Get content pane  
        Container contentPane = getContentPane();  
        contentPane.setLayout(new FlowLayout());  
        // Add radio buttons to content pane  
        JRadioButton b1 = new JRadioButton("A");  
        b1.addActionListener(this);  
        contentPane.add(b1);  
        JRadioButton b2 = new JRadioButton("B");  
        b2.addActionListener(this);  
        contentPane.add(b2);  
        JRadioButton b3 = new JRadioButton("C");  
        b3.addActionListener(this);  
        contentPane.add(b3);  
        // Define a button group
```

```

ButtonGroup bg = new ButtonGroup();
bg.add(b1);
bg.add(b2);
bg.add(b3);
// Create a text field and add it
// to the content pane
tf = new JTextField(5);
contentPane.add(tf);
}

```

a)

```

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

```

b)

```

import java.Applet.*
import java.awt.*;

```

c)

```

import javax.Swing.Tree.*
import javax.Swing.*

```

d)

```

import java.awt.*
Import java.JApplet.*;

```

105) Consider the following output. Find the missing statement in the program.

```

import javax.swing.*;
/* <applet code="JTabbedPaneDemo" width=400 height=100> </applet> */
public class JTabbedPaneDemo extends JApplet

```

```
{
public void init()
{
JTabbedPane jtp = new JTabbedPane();
jtp.addTab("Cities", new CitiesPanel());
jtp.addTab("Colors", new ColorsPanel());
jtp.addTab("Flavors", new FlavorsPanel());
}
}
class CitiesPanel extends JPanel
{
public CitiesPanel()
{
JButton b1 = new JButton("New York");
add(b1);
JButton b2 = new JButton("London");
add(b2);
JButton b3 = new JButton("Hong Kong");
add(b3);
JButton b4 =new JButton("Tokyo"); add(b4);
}
}
class ColorsPanel extends JPanel
{
public ColorsPanel()
{
JCheckBox cb1 = new JCheckBox("Red");
add(cb1);
JCheckBox cb2 = new JCheckBox("Green");
```

```
add(cb2);
JCheckBox cb3 = new JCheckBox("Blue");
add(cb3);
}
}
```

- a) setPane();
- b) getContentPane().add(jtp);**
- c) Both a&b
- d) None of these

106) What is error in following program?

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/* <applet code=ScrollDemo.class width=500 height=500> </applet> */
Public class ScrollDemo extends Applet implements AdjustmentListener
{
    Scrollbar s1,s2,s3;
    public void init(
    {
        s1=new Scrollbar(Scrollbar.VERTICAL,0,1,0,255);
        s2=new Scrollbar(Scrollbar.VERTICAL,0,1,0,255);
        s3=new Scrollbar(Scrollbar.VERTICAL,0,1,0,255);
        add(s1);
        add(s2);
        add(s3);
    }
}
```

```

s1.addAdjustmentListener(this);
s2.addAdjustmentListener(this);
s3.addAdjustmentListener(this);
}
public void adjustmentChange(AdjustmentEvent ae)
{
setBackground(new Color(s1.getValue(),s2.getValue(),s3.getValue()));
}
}

```

- a) ScrollDemo is not abstract and does not override abstract method
- b) adjustmentValueChanged(AdjustmentEvent) in AdjustmentListener public class ScrollDemo extends Applet implements AdjustmentListener
- c) Both a&b**
- d) None of these

107) To get the following output complete the code given below.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class SimpleTableExample extends JFrame
{
private JPanel topPanel;
private JTable table;
private JScrollPane scrollPane;
public SimpleTableExample()
{
setTitle( "Simple Table Application" );

```

```

setSize( 300, 200 );
setBackground( Color.gray );
topPanel = new JPanel();
topPanel.setLayout( new BorderLayout()
);
getContentPane().add( topPanel );
String columnNames[] = { "Column 1", "Column 2","Column 3" };
String dataValues[][] = { { "12", "234", "67" }, { "-123", "43", "853" },
{ "93", "89.2", "109" }, { "279", "9033", "3092" }
};
topPanel.add( scrollPane, BorderLayout.CENTER );
}
public static void main( String args[] )
{
SimpleTableExample mainFrame = new
SimpleTableExample(); mainFrame.setVisible( true );
}
}

```

a)table = new JTable(dataValues,column Names);

scrollPane = new JScrollPane(table);

b)table = new JTable

c)New JScrollPane(t)

d)Both a& b

108) Which statement should be added to display button.

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
/*
```

```
<applet code="JButtonDemo" width=250 height=300>
```

```
</applet>
```



```

*/

public class JButtonDemo extends JApplet
{
public void init()
{
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout());

ImageIcon img= new ImageIcon("jpgIcon.jpg");
JButton jb = new JButton(img);
}
}

```

- a) ContentPane.add(jb);
- b) contentPane.add(jb);**
- c) contentPane.add();
- d) contentPane.addButton(jb);

109) What will be the output of following code:

```

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

/*<applet code="JTreeEvents" width=400 height=200></applet>*/

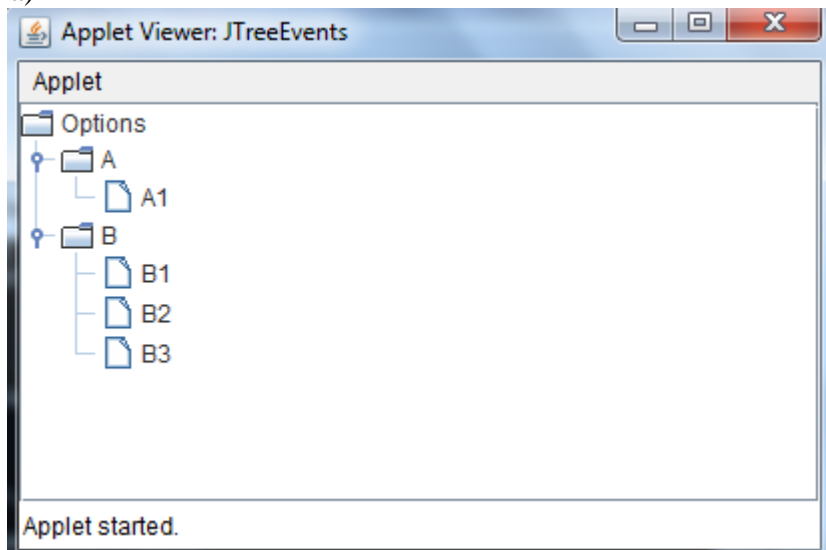
public class JTreeEvents extends JApplet
{
JTree tree;

public void init()
{
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout());
DefaultMutableTreeNode top = new DefaultMutableTreeNode("Options");
DefaultMutableTreeNode a = new DefaultMutableTreeNode("A");
top.add(a);
DefaultMutableTreeNode a1 = new DefaultMutableTreeNode("A1");
a.add(a1);
DefaultMutableTreeNode a2 = new DefaultMutableTreeNode("A2");
a.add(a2);
DefaultMutableTreeNode b = new DefaultMutableTreeNode("B");
top.add(b);
DefaultMutableTreeNode b1 = new DefaultMutableTreeNode("B1");
b.add(b1);
DefaultMutableTreeNode b2 = new DefaultMutableTreeNode("B2");
b.add(b2);
DefaultMutableTreeNode b3 = new DefaultMutableTreeNode("B3");

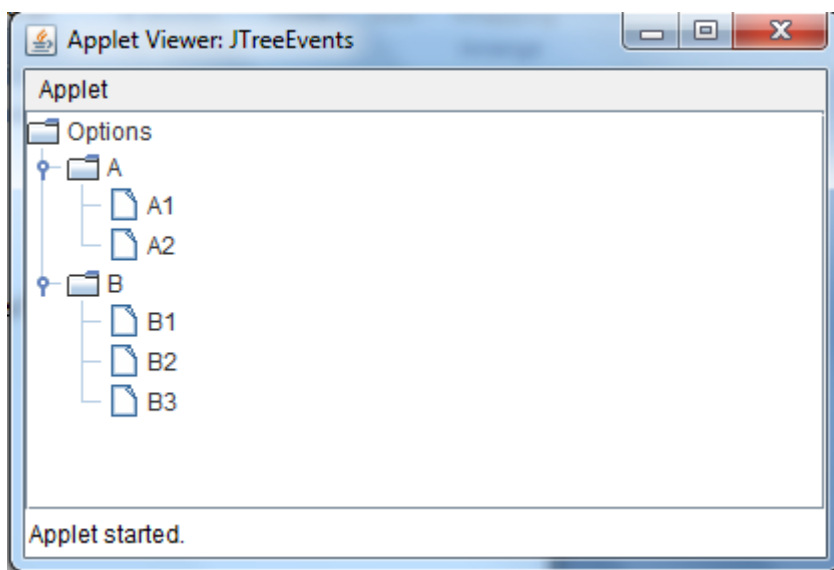
```

```
b.add(b3);
tree = new JTree(top);
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPane jsp = new JScrollPane(tree, v, h);
contentPane.add(jsp, BorderLayout.CENTER);
}
}
```

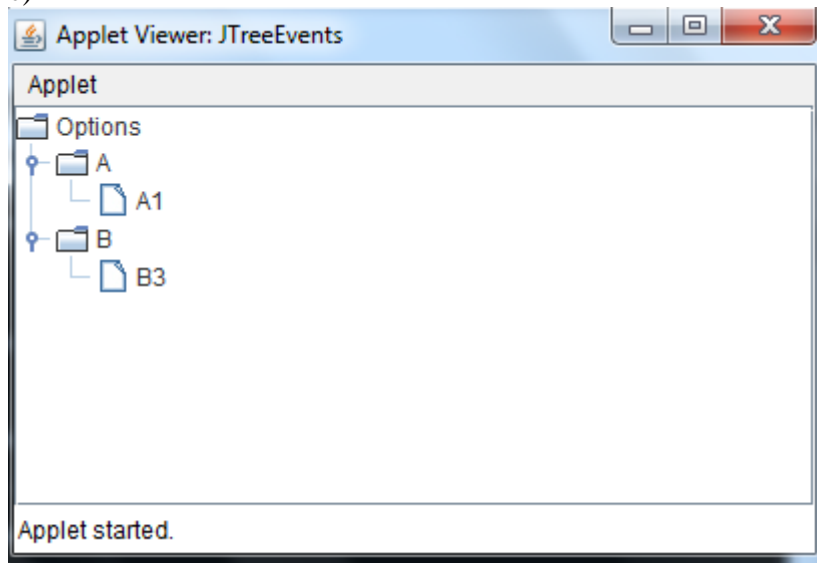
a)



b)



c)



d)

