**1)Write a programme to demonstrate the use of border layout which shows four buttons at four sides of an applet with captions as East, West, North, South.**

**Ans:-**

import java.awt.\*;

import java.applet.\*;

public class NewApplet1 extends Applet

{

public void init() {

Button btnEast = new Button("East");

Button btnWest = new Button("West");

Button btnNorth = new Button("North");

Button btnSouth = new Button("South");

setLayout(new BorderLayout());

add(btnEast, BorderLayout.EAST);

add(btnWest, BorderLayout.WEST);

add(btnNorth, BorderLayout.NORTH);

add(btnSouth, BorderLayout.SOUTH);

}

}

/\* <applet code="Question1.class" width=500 height=500></applet>\*/

**2) Write program to create an applet that will accept values of three numbers the user will enter the values in three separate text fields the Applet will have two buttons labelled find largest and find smallest when the user will click on button find largest the largest value among the 3 number will be displayed in 4th text field and when the user will click on the button find smallest smallest value among three numbers will be displayed in 4th text field.**

**Ans:-**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.applet.\*;

public class NewApplet1 extends Applet implements ActionListener {

TextField t1, t2, t3, t4;

Button b1, b2;

public void init() {

t1 = new TextField();

t2 = new TextField();

t3 = new TextField();

t4 = new TextField();

b1 = new Button("Largest");

b2 = new Button("Smallest");

t4.setEditable(false);

b1.addActionListener(this);

b2.addActionListener(this);

setLayout(new FlowLayout());

add(t1);

add(t2);

add(t3);

add(t4);

add(b1);

add(b2);

}

public void actionPerformed(ActionEvent e) {

Integer num1 = Integer.parseInt(t1.getText());

Integer num2 = Integer.parseInt(t2.getText());

Integer num3 = Integer.parseInt(t3.getText());

if (e.getSource() == b1) {

if (num1 > num2 && num1 > num3) {

t4.setText(num1.toString());

} else if (num2 > num1 && num2 > num3) {

t4.setText(num2.toString());

} else {

t4.setText(num3.toString());

}

} else if (e.getSource() == b2) {

if (num1 < num2 && num1 < num3) {

t4.setText(num1.toString());

} else if (num2 < num1 && num2 < num3) {

t4.setText(num2.toString());

} else {

t4.setText(num3.toString());

}

}

}

}

/\* <applet code="Question2.class" width=500 height=500></applet>\*/

**3) Write a programme to create an applet that will accept a number in text file and display factorial of that number in another text field when the button caption factorial is clicked.**

**Ans:-**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.applet.\*;

public class NewApplet1 extends Applet implements ActionListener {

TextField t1, t2;

Button b1;

public void init() {

t1 = new TextField();

t2 = new TextField();

b1 = new Button("Factorial");

b1.addActionListener(this);

setLayout(new FlowLayout());

add(t1);

add(t2);

add(b1);

}

public void actionPerformed(ActionEvent e)

{

Integer num1 = Integer.parseInt(t1.getText());

Integer fact=1;

for(int i=1;i<=num1;i++)

{

fact=fact\*i;

}

t2.setText(fact.toString());

}

}

/\* <applet code="Question3.class" width=500 height=500></applet>\*/

**4) Write a programme to create a frame which includes two labels namely username and password 2 text field and 1 button with caption submit**

**Ans:-**

import javax.swing.\*;

import java.awt.\*;

public class NewClass extends Frame

{

Label l1,l2;

TextField t1,t2;

Button b1;

public NewClass()

{

l1 = new Label("Username:");

l2 = new Label("Password:");

t1 = new TextField();

t2 = new TextField();

b1 = new Button("Submit");

add(l1);

add(l2);

add(t1);

add(t2);

add(b1);

setVisible(true);

setSize(500,500);

setLayout(null);

l1.setBounds(50, 50, 80, 30);

t1.setBounds(140, 50, 120, 30);

l2.setBounds(50, 100, 80, 30);

t2.setBounds(140, 100, 120, 30);

b1.setBounds(100, 150, 80, 30);

}

public static void main(String args[])

{

new NewClass();

}

}

**5) Write a programme to create an applet which will accept two numbers from user and will provide result of four operations addition subtraction multiplication and division on selection of any of these items operation provided in one choice box.**

**Ans:-**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class NewApplet1 extends Applet implements ActionListener {

TextField t1, t2, resultField;

Choice c1;

public void init() {

Label l1 = new Label("Enter number 1:");

Label l2 = new Label("Enter number 2:");

Label l3 = new Label("Select operation:");

t1 = new TextField(10);

t2 = new TextField(10);

resultField = new TextField(10);

resultField.setEditable(false);

c1 = new Choice();

c1.add("Addition");

c1.add("Subtraction");

c1.add("Multiplication");

c1.add("Division");

Button b1 = new Button("Calculate");

b1.addActionListener(this);

setLayout(new GridLayout(4, 2));

add(l1);

add(t1);

add(l2);

add(t2);

add(l3);

add(c1);

add(b1);

add(resultField);

}

public void actionPerformed(ActionEvent e) {

double num1 = Double.parseDouble(t1.getText());

double num2 = Double.parseDouble(t2.getText());

String operation = c1.getSelectedItem();

double result = 0;

switch (operation) {

case "Addition":

result = num1 + num2;

break;

case "Subtraction":

result = num1 - num2;

break;

case "Multiplication":

result = num1 \* num2;

break;

case "Division":

result = num1 / num2;

break;

}

resultField.setText(String.valueOf(result));

}

}

/\* <applet code="Question5.class" width=500 height=500></applet>\*/

**6) Write a programme to create three menus on the frame add menu item Checkbox menu item and menu shortcut to it**

**Ans:-**

import javax.swing.\*;

import java.awt.event.\*;

public class NewApplet1 {

public static void main(String[] args) {

JFrame frame = new JFrame("Simple Menu Example");

frame.setSize(300, 200);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JMenuBar menuBar = new JMenuBar();

JMenu jm1 = new JMenu("Add Items");

jm1.add(new JMenuItem("Item 1"));

jm1.add(new JMenuItem("Item 2"));

jm1.add(new JMenuItem("Item 3"));

JMenu jm2 = new JMenu("Checkbox MenuItems");

jm2.add(new JCheckBoxMenuItem("Checkbox 1"));

jm2.add(new JCheckBoxMenuItem("Checkbox 2"));

jm2.add(new JCheckBoxMenuItem("Checkbox 3"));

JMenu jm3 = new JMenu("Menu Shortcuts");

jm3.add(new JMenuItem("Shortcut 1"));

jm3.add(new JMenuItem("Shortcut 2"));

jm3.add(new JMenuItem("Shortcut 3"));

menuBar.add(jm1);

menuBar.add(jm2);

menuBar.add(jm3);

frame.setJMenuBar(menuBar);

frame.setVisible(true);

}

}

**7) Write a program with a AWT Component List and demonstrate the use of following**

**methods on the component.**

**1) getitem()**

**ii) add()**

**3)getItemCount()**

**iv) select()**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

private List componentList;

public ListExample() {

setTitle("AWT List Example");

setSize(300, 200);

setLayout(new FlowLayout());

componentList = new List();

componentList.add("Item 1");

componentList.add("Item 2");

componentList.add("Item 3");

add(componentList);

Button selectButton = new Button("Select Item");

selectButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

String selectedItem = componentList.getItem(1);

componentList.select(1);

System.out.println("Selected Item: " + selectedItem);

}

});

add(selectButton);

Button addButton = new Button("Add Item");

addButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

int itemCount = componentList.getItemCount();

componentList.add("New Item " + (itemCount + 1));

System.out.println("Item Added. New Item Count: " + componentList.getItemCount());

}

});

add(addButton);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public static void main(String[] args) {

ListExample example = new ListExample();

example.setVisible(true);

}

}

**8) Write a programme to design a calculator with grid bag layout.**

**Ans:-**

import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import javax.swing.\*;

public class ListExample extends JFrame {

ListExample() {

GridBagConstraints c = new GridBagConstraints();

setLayout(new GridBagLayout());

JTextField t1 = new JTextField();

JButton btn0 = new JButton("0");

JButton btn1 = new JButton("1");

JButton btn2 = new JButton("2");

JButton btn3 = new JButton("3");

JButton btn4 = new JButton("4");

JButton btn5 = new JButton("5");

JButton btn6 = new JButton("6");

JButton btn7 = new JButton("7");

JButton btn8 = new JButton("8");

JButton btn9 = new JButton("9");

JButton btnadd = new JButton("+");

JButton btnsub = new JButton("-");

JButton btnmul = new JButton("\*");

JButton btndiv = new JButton("/");

JButton btneq = new JButton("=");

JButton btnclr = new JButton("C");

c.gridx = 0;

c.gridy = 0;

c.gridwidth = 4; // Span two columns

c.fill = GridBagConstraints.HORIZONTAL;

add(t1, c);

c.gridwidth = 1;

c.gridx = 0;

c.gridy = 1;

add(btn7, c);

c.gridx = 1;

add(btn8, c);

c.gridx = 2;

add(btn9, c);

c.gridx = 3;

add(btnclr, c);

c.gridx = 0;

c.gridy = 2;

add(btn4, c);

c.gridx = 1;

add(btn5, c);

c.gridx = 2;

add(btn6, c);

c.gridx = 3;

add(btnmul, c);

c.gridx = 0;

c.gridy = 3;

add(btn1, c);

c.gridx = 1;

add(btn2, c);

c.gridx = 2;

add(btn3, c);

c.gridx = 3;

add(btndiv, c);

c.gridy = 4;

c.gridx = 0;

add(btneq, c);

c.gridx = 1;

add(btn0, c);

c.gridx = 2;

add(btnsub, c);

c.gridx = 3;

add(btnadd, c);

setVisible(true);

setSize(500, 500);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public static void main(String[] args) {

new ListExample();

}

}

**Or**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

TextField t1;

public ListExample() {

Panel p2 = new Panel();

t1 = new TextField();

Button[] buttons = new Button[]{

new Button("7"), new Button("8"), new Button("9"), new Button("\*"),

new Button("4"), new Button("5"), new Button("6"), new Button("/"),

new Button("1"), new Button("2"), new Button("3"), new Button("-"),

new Button("0"), new Button("."), new Button("="), new Button("+")

};

setLayout(new BorderLayout());

setSize(500, 500);

add(t1, BorderLayout.NORTH);

p2.setLayout(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.fill = GridBagConstraints.BOTH;

gbc.weightx = 1.0;

gbc.weighty = 1.0;

int row = 0;

int col = 0;

for (Button button : buttons) {

gbc.gridx = col;

gbc.gridy = row;

p2.add(button, gbc);

col++;

if (col > 3) {

col = 0;

row++;

}

}

add(p2, BorderLayout.CENTER);

setVisible(true);

}

public static void main(String args[]) {

new ListExample();

}

}

**9) Write a program to display J progress bar to show progress of task.**

**Ans:-**

import javax.swing.\*;

import java.lang.\*;

public class ListExample extends JFrame

{

JProgressBar jb;

int i=0 ;

ListExample()

{

jb = new JProgressBar(0,200);

jb.setBounds(40,40,200,30);

jb.setValue(0);

jb.setStringPainted(true);

add(jb);

setSize(500,500);

setLayout(null);

setVisible(true);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void color()

{

while( i<=2000)

{

jb.setValue(i);

i=i+20;

try

{

Thread.sleep(150);

}

catch(Exception e)

{

System.out.println(e);

}

}

}

public static void main(String args[])

{

ListExample q1 = new ListExample();

q1.color();

}

}

**10) Write a program to create an applet that will accept values of three test marks i.e test 1 test 2 test 3 each out of 25 the user will enter the marks in three separate text fields the applet will have a button labelled find average when the user will click on the button the average of test marks will be displayed on the 4th text field.**

**Ans:-**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class NewApplet1 extends Applet implements ActionListener {

TextField t1,t2,t3,t4;

Button b1;

public void init() {

t1 = new TextField();

t2 = new TextField();

t3 = new TextField();

t4 = new TextField();

b1 = new Button("Find Avegrage");

b1.addActionListener(this);

setLayout(new FlowLayout());

add(t1);

add(t2);

add(t3);

add(t4);

add(b1);

}

public void actionPerformed(ActionEvent e)

{

Integer num1 = Integer.parseInt(t1.getText());

Integer num2 = Integer.parseInt(t2.getText());

Integer num3 = Integer.parseInt(t3.getText());

Integer avg = (num1+num2+num3)/3;

t4.setText(avg.toString());

}

}

/\* <applet code="Question10.class" width=500 height=500></applet>\*/

**11) Write a programme to demonstrate the use of key on applet windows such as key pressed key released key typed.**

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class NewApplet1 extends Applet implements KeyListener

{

TextArea t1;

Label l1;

public void init()

{

t1 = new TextArea();

t1.setBounds(20,80,200,250);

l1 = new Label();

l1.setBounds(20,50,100,20);

t1.addKeyListener(this);

add(t1);

add(l1);

setLayout(null);

}

public void keyPressed(KeyEvent e)

{

l1.setText("Key Pressed");

}

public void keyTyped(KeyEvent e)

{

l1.setText("Key Typed");

}

public void keyReleased(KeyEvent e)

{

l1.setText("Key Released");

}

}

/\* <applet code="Question11.class" width=500 height=500></applet>\*/

**12) Write a programme to create an applet that will accept a number in text field and display square of that number in another text field when a button with caption Square is clicked.**

**Ans:-**

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

public class NewApplet1 extends Applet implements ActionListener {

TextField t1, t2;

Button b1;

public void init() {

t1 = new TextField();

t2 = new TextField();

b1 = new Button("Sqaure");

b1.addActionListener(this);

setLayout(new FlowLayout());

add(t1);

add(t2);

add(b1);

}

public void actionPerformed(ActionEvent e)

{

Integer num1 = Integer.parseInt(t1.getText());

Integer res = num1\*num1;

t2.setText(res.toString());

}

}

/\* <applet code="Question12.class" width=500 height=500></applet>\*/

**13) Write a programme to demonstrate the use of checkbox and checkboxgroup class.**

**Ans:-**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

public ListExample() {

setTitle("Checkbox Example");

setSize(300, 200);

setLayout(new FlowLayout());

setVisible(true);

Label l1=new Label("Gender:");

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

CheckboxGroup c1 = new CheckboxGroup();

Checkbox cb1 = new Checkbox("Male", c1, false);

Checkbox cb2 = new Checkbox("Female", c1, false);

Checkbox cb3 = new Checkbox("Other", c1, false);

add(l1);

add(cb1);

add(cb2);

add(cb3);

}

public static void main(String[] args) {

new ListExample();

}

}

**13) Write a programme to demonstrate the use of checkbox and checkboxgroup class.**

**Ans:-**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

public ListExample() {

setTitle("Checkbox Example");

setSize(300, 200);

setLayout(new FlowLayout());

setVisible(true);

Label l1=new Label("Gender:");

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

CheckboxGroup c1 = new CheckboxGroup();

Checkbox cb1 = new Checkbox("Male", c1, false);

Checkbox cb2 = new Checkbox("Female", c1, false);

Checkbox cb3 = new Checkbox("Other", c1, false);

Checkbox cssCheckbox = new Checkbox("CSS");

Checkbox ajpCheckbox = new Checkbox("AJP");

add(cssCheckbox);

add(ajpCheckbox);

add(l1);

add(cb1);

add(cb2);

add(cb3);

}

public static void main(String[] args) {

new ListExample();

}

}

**14) Write a program with a AWT Component Choice and demonstrate the use of following methods on the component.**

**i) getitemCount()**

**ii) select()**

**iii) getitem()**

**iv) add()**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

private Choice componentChoice;

public ListExample() {

setTitle("AWT Choice Example");

setSize(300, 200);

setLayout(new FlowLayout());

componentChoice = new Choice();

componentChoice.add("Item 1");

componentChoice.add("Item 2");

componentChoice.add("Item 3");

add(componentChoice);

Button selectButton = new Button("Select Item");

selectButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int itemCount = componentChoice.getItemCount();

if (itemCount > 0) {

int selectedIndex = (int) (Math.random() \* itemCount);

componentChoice.select(selectedIndex);

System.out.println("Selected Item: " + componentChoice.getItem(selectedIndex));

} else {

System.out.println("Choice is empty.");

}

}

});

add(selectButton);

Button addButton = new Button("Add Item");

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

int itemCount = componentChoice.getItemCount();

componentChoice.add("New Item " + (itemCount + 1));

System.out.println("Item Added. New Item Count: " + componentChoice.getItemCount());

}

});

add(addButton);

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public static void main(String[] args) {

ListExample example = new ListExample();

example.setVisible(true);

}

}

**14) Write a programme to display J Combo box on Applet which have following items blue Red Green Yellow Pink when users select any of the item the selected item will get displayed on applet**

**Ans:-**

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

public class ListExample extends MouseAdapter

{

JComboBox jcb;

Label l1;

ListExample()

{

JFrame f = new JFrame();

String[] data={"Blue","Red","Green","Yellow"};

jcb = new JComboBox(data);

l1 = new Label("Null");

f.setLayout(new BorderLayout());

f.add(jcb, BorderLayout.NORTH);

f.add(l1, BorderLayout.CENTER);

jcb.addMouseListener(this);

f.setVisible(true);

f.setSize(500,500);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

public void mouseClicked(MouseEvent e)

{

l1.setText(jcb.getSelectedItem().toString());

}

public static void main(String[] args)

{

new ListExample();

}

}

**15) Write a programme demonstrate mouse event using mouse adapter class.**

**Ans:-**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

Label label;

public ListExample() {

// Frame setup

setTitle("Mouse Event Demo");

setSize(400, 300);

setLayout(new FlowLayout());

// Label to display messages

label = new Label("Perform a mouse action...");

add(label);

// Add MouseAdapter for mouse events

addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

label.setText("Mouse Clicked at (" + e.getX() + ", " + e.getY() + ")");

}

public void mouseEntered(MouseEvent e) {

label.setText("Mouse Entered the frame");

}

public void mouseExited(MouseEvent e) {

label.setText("Mouse Exited the frame");

}

});

// Window close event

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent we) {

System.exit(0);

}

});

setVisible(true);

}

public static void main(String[] args) {

new ListExample();

}

}

**16) Write a programme to display Ajay Kumar box on Apple at the combo box will have the following items banana Apple Orange Chiku Crepes when user selects an item from it the selected item and it’s selected index will be displayed in the text field.**

**Ans:-**

import java.applet.Applet;

import java.awt.\*;

import java.awt.event.\*;

/\*

<applet code="ComboBoxApplet" width=400 height=300>

</applet>

\*/

public class NewApplet1 extends Applet implements ItemListener {

Choice comboBox; // Combo box

TextField textField; // Text field to display selection

public void init() {

// Set layout

setLayout(new FlowLayout());

// Add a label

Label label = new Label("Ajay Kumar Box:");

add(label);

// Create a combo box (Choice)

comboBox = new Choice();

comboBox.add("Banana");

comboBox.add("Apple");

comboBox.add("Orange");

comboBox.add("Chiku");

comboBox.add("Grapes");

add(comboBox);

// Add a text field for displaying selected item and index

textField = new TextField(30);

textField.setEditable(false);

add(textField);

// Add item listener to the combo box

comboBox.addItemListener(this);

}

// Handle item selection

public void itemStateChanged(ItemEvent e) {

String selectedItem = comboBox.getSelectedItem();

int selectedIndex = comboBox.getSelectedIndex();

textField.setText("Selected Item: " + selectedItem + ", Index: " + selectedIndex);

}

}

**17) Write a programme to display J3 in Applette when user selects any node from J3 it path from root node should be displayed in the text field.  
Ans:-**

import javax.swing.\*;

import javax.swing.event.TreeSelectionEvent;

import javax.swing.event.TreeSelectionListener;

import javax.swing.tree.DefaultMutableTreeNode;

import javax.swing.tree.TreePath;

import java.awt.\*;

public class NewJApplet1 extends JApplet {

private JTree tree;

private JTextField pathTextField;

public void init() {

// Create the root node and add child nodes

DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");

DefaultMutableTreeNode node1 = new DefaultMutableTreeNode("Node 1");

DefaultMutableTreeNode node2 = new DefaultMutableTreeNode("Node 2");

DefaultMutableTreeNode node3 = new DefaultMutableTreeNode("Node 3");

root.add(node1);

root.add(node2);

node2.add(node3);

// Create the JTree with the root node

tree = new JTree(root);

// Add a TreeSelectionListener to handle node selection

tree.addTreeSelectionListener(new TreeSelectionListener() {

public void valueChanged(TreeSelectionEvent e) {

displaySelectedPath();

}

});

// Create a JTextField for displaying the selected path

pathTextField = new JTextField();

pathTextField.setEditable(false);

// Set layout manager to BorderLayout

setLayout(new BorderLayout());

// Add the JTree directly to the applet's content pane

add(tree, BorderLayout.CENTER);

// Add the JTextField to the applet's content pane

add(pathTextField, BorderLayout.SOUTH);

}

private void displaySelectedPath() {

// Get the selected path

TreePath selectedPath = tree.getSelectionPath();

if (selectedPath != null) {

// Convert the path to a string and display it in the text field

String path = "";

Object[] pathComponents = selectedPath.getPath();

for (Object component : pathComponents) {

path += component.toString() + " -> ";

}

path = path.substring(0, path.length() - 4); // Remove the last " -> "

pathTextField.setText("Selected Path: " + path);

} else {

pathTextField.setText("No node selected");

}

}

}

**18) Write a program to create a frame using swing with sitte changing colors having buttons Red, Green, Blue Clicking on this button background should change to**

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class ListExample extends JFrame implements ActionListener {

private JPanel colorPanel;

public ListExample() {

// Set up the frame

setTitle("Color Changing Frame");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create a panel to change its background color

colorPanel = new JPanel();

add(colorPanel);

// Create buttons

JButton redButton = new JButton("Red");

JButton greenButton = new JButton("Green");

JButton blueButton = new JButton("Blue");

// Add action listeners to buttons

redButton.addActionListener(this);

greenButton.addActionListener(this);

blueButton.addActionListener(this);

// Create a panel for buttons

JPanel buttonPanel = new JPanel();

buttonPanel.add(redButton);

buttonPanel.add(greenButton);

buttonPanel.add(blueButton);

// Add button panel to the frame

add(buttonPanel, "South");

// Set default background color

colorPanel.setBackground(Color.RED);

}

public void actionPerformed(ActionEvent e) {

String command = e.getActionCommand();

switch (command) {

case "Red":

colorPanel.setBackground(Color.RED);

break;

case "Green":

colorPanel.setBackground(Color.GREEN);

break;

case "Blue":

colorPanel.setBackground(Color.BLUE);

break;

}

}

public static void main(String[] args) {

// Create and show the frame

javax.swing.SwingUtilities.invokeLater(new Runnable() {

public void run() {

new ListExample().setVisible(true);

}

});

}

}

**19) Write a program to display a button with an image on it using swing. The image on the button will change its image when the it is clicked:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ImageIcon;

public class ListExample extends JFrame implements ActionListener {

private JButton imageButton;

private ImageIcon firstImage, secondImage;

public ListExample() {

// Set up the frame

setTitle("Image Changing Button");

setSize(300, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create images

firstImage = new ImageIcon("C:\\Users\\hp\\OneDrive\\Pictures\\reset.jpeg");

secondImage = new ImageIcon("C:\\Users\\hp\\OneDrive\\Pictures\\Screenshot 2024-07-20 164542.png");

// Create a button with the initial image

imageButton = new JButton(firstImage);

imageButton.addActionListener(this);

// Add button to the frame

add(imageButton, BorderLayout.CENTER);

}

public void actionPerformed(ActionEvent e) {

// Change the image on the button when clicked

if (e.getSource() == imageButton) {

if (imageButton.getIcon().equals(firstImage)) {

imageButton.setIcon(secondImage);

} else {

imageButton.setIcon(firstImage);

}

}

}

public static void main(String[] args) {

// Create and show the frame

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new ListExample().setVisible(true);

}

});

}

}

**20) Write a program to display an applet using swing. The applet contains one JLabel, one Image Icon & one JButton it also applet contains image as background of applet.**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class ListExample {

private JFrame frame;

private JLabel label;

private ImageIcon imageIcon;

private JButton button;

public ListExample () {

// Set up the frame

frame = new JFrame("Swing Applet Example");

frame.setSize(400, 300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create a label

label = new JLabel("Hello, Swing!", JLabel.CENTER);

// Load an image and create an ImageIcon

imageIcon = new ImageIcon("C:\\Users\\hp\\OneDrive\\Pictures\\Screenshot 2024-07-20 164542.png");

// Create a button

button = new JButton("Click Me");

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Change label text on button click

label.setText("Button Clicked!");

}

});

// Set the background image

JPanel backgroundPanel = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

g.drawImage(imageIcon.getImage(), 0, 0, getWidth(), getHeight(), this);

}

};

// Set layout manager

backgroundPanel.setLayout(new BorderLayout());

// Add components to the background panel

backgroundPanel.add(label, BorderLayout.CENTER);

backgroundPanel.add(button, BorderLayout.SOUTH);

// Set the content pane of the frame

frame.setContentPane(backgroundPanel);

// Center the frame on the screen

frame.setLocationRelativeTo(null);

}

public void display() {

// Make the frame visible

frame.setVisible(true);

}

public static void main(String[] args) {

// Create and show the SwingAppletExample

ListExample appletExample = new ListExample ();

appletExample.display();

}

}

**21) Write a program to create and display JTable on Applet**

import javax.swing.\*;

public class ListExample extends JFrame {

ListExample() {

String[][] data = {

{"101","John","69000"},

{"102","Chris","96000"}};

String[] col = {"ID", "NAME", "SALARY"};

JTable jt = new JTable(data,col);

JScrollPane jScrollPane = new JScrollPane(jt);

setLayout(null);

jScrollPane.setBounds(30,40,200,300);

add(jScrollPane);

setSize(300,300);

setVisible(true);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public static void main(String[] args) {

new ListExample();

}

}

**22) Write a program to display JTree in the applet. When user selects any node from JTree its path(from root node) should be displayed in the TextField. Add Jtree in JScrollPane**

import javax.swing.\*;

import javax.swing.event.TreeSelectionEvent;

import javax.swing.event.TreeSelectionListener;

import javax.swing.tree.DefaultMutableTreeNode;

import javax.swing.tree.TreePath;

import java.awt.\*;

public class NewJApplet extends JApplet {

private JTree tree;

private JTextField pathTextField;

public void init() {

// Create the root node and add child nodes

DefaultMutableTreeNode root = new DefaultMutableTreeNode("Root");

DefaultMutableTreeNode node1 = new DefaultMutableTreeNode("Node 1");

DefaultMutableTreeNode node2 = new DefaultMutableTreeNode("Node 2");

DefaultMutableTreeNode node3 = new DefaultMutableTreeNode("Node 3");

root.add(node1);

root.add(node2);

node2.add(node3);

// Create the JTree with the root node

tree = new JTree(root);

// Add a TreeSelectionListener to handle node selection

tree.addTreeSelectionListener(new TreeSelectionListener() {

public void valueChanged(TreeSelectionEvent e) {

displaySelectedPath();

}

});

// Create a JScrollPane and add the JTree to it

JScrollPane treeScrollPane = new JScrollPane(tree);

// Create a JTextField for displaying the selected path

pathTextField = new JTextField();

pathTextField.setEditable(false);

// Set layout manager to BorderLayout

setLayout(new BorderLayout());

// Add components to the applet

add(treeScrollPane, BorderLayout.CENTER);

add(pathTextField, BorderLayout.SOUTH);

}

private void displaySelectedPath() {

// Get the selected path

TreePath selectedPath = tree.getSelectionPath();

if (selectedPath != null) {

// Convert the path to a string and display it in the text field

String path = "";

Object[] pathComponents = selectedPath.getPath();

for (Object component : pathComponents) {

path += component.toString() + " -> ";

}

path = path.substring(0, path.length() - 4); // Remove the last " -> "

pathTextField.setText("Selected Path: " + path);

} else {

pathTextField.setText("No node selected");

}

}

}

**25) Write a Program to demonstrate the use of InetAddress class along with its Factory and Instance methods**

import java.net.\*;

public class ListExample {

public static void main(String[] args)throws Exception {

InetAddress inet=InetAddress.getLocalHost();

System.out.println("Local Host:"+inet);

System.out.println("Local HostName:"+inet.getHostName());

System.out.println("Local HostAddress(:"+inet.getHostAddress());

System.out.println("Local CanonicalHostName(:"+inet.getCanonicalHostName());

InetAddress inet1=InetAddress.getByName("www.google.com");

System.out.println("Google Host:"+inet1);

System.out.println("Google HostName:"+inet1.getHostName());

System.out.println("Google HostAddress(:"+inet1.getHostAddress());

InetAddress inet3[]=InetAddress.getAllByName("www.yahoo.com");

for(int i=0 ; i<inet3.length ;i++)

{

System.out.println(inet3[i]);

}

}

}

**26) Write a Program to demonstrate the use of URL and URLConnection class.**

import java.io.DataInputStream;

import java.net.\*;

public class ListExample {

public static void main(String[] args)throws Exception {

URL url=new URL("https://msbte.ac.in/");

System.out.println("Protocol:"+url.getProtocol());

System.out.println("Port:"+url.getPort());

System.out.println("Domain Name:"+url.getHost());

System.out.println("File Name:"+url.getFile());

URLConnection u=url.openConnection();

DataInputStream din=new DataInputStream(u.getInputStream());

int i;

while((i=din.read())!=-1)

{

System.out.print((char)i);

}

}

}

import java.io.DataInputStream;

import java.net.\*;

import java.util.Date;

public class url {

public static void main(String[] args)throws Exception {

URL url=new URL("https://msbte.ac.in/");

URLConnection uc=url.openConnection();

System.out.println("Date:"+new Date(uc.getDate()));

System.out.println("Content Type:"+uc.getContentType());

System.out.println("Content Type length:"+uc.getContentLength());

}

}