**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 Question1 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 Question2 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.seytEditable(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 Question3 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 Question4 extends Frame

{

Label l1,l2;

TextField t1,t2;

Button b1;

public Question4()

{

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(50,50,80,80);

t1.setBounds(130,50,60,20);

l2.setBounds(50,110,80,80);

t2.setBounds(130,110,60,20);

b1.setBounds(50,150,30,30);

}

public static void main(String args[])

{

new Question4();

}

}

**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 Question5 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 Question6 {

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()**

**getItemCount()**

**iv) select()**

import java.awt.\*;

import java.awt.event.\*;

public class ListExample extends Frame {

private List componentList;

public ListExample() {

// Set frame properties

setTitle("AWT List Example");

setSize(300, 200);

setLayout(new FlowLayout());

// Create a List component

componentList = new List();

// Add items to the List

componentList.add("Item 1");

componentList.add("Item 2");

componentList.add("Item 3");

// Add List component to the frame

add(componentList);

// Add a Button to demonstrate selecting an item

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

selectButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Demonstrate getItem() and select() methods

String selectedItem = componentList.getItem(1); // Get the item at index 1

componentList.select(1); // Select the item at index 1

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

}

});

add(selectButton);

// Add a Button to demonstrate getItemCount() and add() methods

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

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Demonstrate getItemCount() and add() methods

int itemCount = componentList.getItemCount(); // Get the current item count

componentList.add("New Item " + (itemCount + 1)); // Add a new item

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

}

});

add(addButton);

// Add a WindowListener to handle window closing event

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.\*;

import java.awt.event.\*;

public class Question8 extends Frame {

TextField t1;

public Question8() {

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 Question8();

}

}

**OR**

import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import javax.swing.\*;

public class GridBagLayoutDemo extends JFrame {

GridBagLayoutDemo() {

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 GridBagLayoutDemo();

}

}

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

**Ans:-**

import javax.swing.\*;

import java.lang.\*;

public class Question9 extends JFrame

{

JProgressBar jb;

int i=0 ;

Question9()

{

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[])

{

Question9 q1 = new Question9();

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 Question10 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.**

Ans:-

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Question11 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 Question12 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 Question13 extends Frame {

public Question13() {

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 Question13();

}

}

**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 ChoiceExample extends Frame {

private Choice componentChoice;

public ChoiceExample() {

// Set frame properties

setTitle("AWT Choice Example");

setSize(300, 200);

setLayout(new FlowLayout());

// Create a Choice component

componentChoice = new Choice();

// Add items to the Choice

componentChoice.add("Item 1");

componentChoice.add("Item 2");

componentChoice.add("Item 3");

// Add Choice component to the frame

add(componentChoice);

// Add a Button to demonstrate selecting an item

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

selectButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Demonstrate getItemCount() and select() methods

int itemCount = componentChoice.getItemCount(); // Get the current item count

if (itemCount > 0) {

int selectedIndex = (int) (Math.random() \* itemCount); // Randomly select an item

componentChoice.select(selectedIndex); // Select the item at the random index

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

} else {

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

}

}

});

add(selectButton);

// Add a Button to demonstrate getItemCount() and add() methods

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

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Demonstrate getItemCount() and add() methods

int itemCount = componentChoice.getItemCount(); // Get the current item count

componentChoice.add("New Item " + (itemCount + 1)); // Add a new item

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

}

});

add(addButton);

// Add a WindowListener to handle window closing event

addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

}

public static void main(String[] args) {

ChoiceExample example = new ChoiceExample();

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 Question14 extends MouseAdapter

{

JComboBox jcb;

Label l1;

public Question14()

{

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 Question14();

}

}

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

**Ans:-**

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

public class Question15 extends MouseAdapter

{

JComboBox jcb;

Label l1;

public Question15()

{

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 Question15();

}

}

**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.awt.event.\*;

import java.awt.\*;

import javax.swing.\*;

public class Question16 extends MouseAdapter

{

JComboBox jcb;

TextField t1;

public Question16()

{

JFrame f = new JFrame();

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

jcb = new JComboBox(data);

t1 = new TextField();

f.setLayout(new BorderLayout());

f.add(jcb, BorderLayout.NORTH);

f.add(t1, BorderLayout.SOUTH);

jcb.addMouseListener(this);

f.setVisible(true);

f.setSize(500,500);

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

}

public void mouseClicked(MouseEvent e)

{

t1.setText(jcb.getSelectedItem().toString());

}

public static void main(String[] args)

{

new Question16();

}

}

**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 JTreeApplet 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 ColorChangingFrame extends JFrame implements ActionListener {

private JPanel colorPanel;

public ColorChangingFrame() {

// 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 ColorChangingFrame().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 ImageChangingButton extends JFrame implements ActionListener {

private JButton imageButton;

private ImageIcon firstImage, secondImage;

public ImageChangingButton() {

// Set up the frame

setTitle("Image Changing Button");

setSize(300, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// Create images

firstImage = new ImageIcon("path/to/firstImage.png");

secondImage = new ImageIcon("path/to/secondImage.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 ImageChangingButton().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 SwingAppletExample {

private JFrame frame;

private JLabel label;

private ImageIcon imageIcon;

private JButton button;

public SwingAppletExample() {

// 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("path/to/your/image.jpg");

// 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

SwingAppletExample appletExample = new SwingAppletExample();

appletExample.display();

}

}

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

import javax.swing.\*;

public class JTableDemo extends JFrame {

JTableDemo() {

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 JTableDemo();

}

}

**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 JTreeApplet 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");

}

}

}

**23) Write a program to create Employee Database. Write the following SQL queries**

**a) Create Table Employee\_Records (Assume suitable data)**

**b) Insert 3 Records in the table**

**c) Display all Records**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class EmployeeDatabase {

private static final String JDBC\_URL = "jdbc:mysql://localhost:3306/your\_database\_name";

private static final String USER = "your\_username";

private static final String PASSWORD = "your\_password";

public static void main(String[] args) {

createTable();

insertRecords();

displayRecords();

}

private static void createTable() {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

Statement statement = connection.createStatement()) {

// SQL query to create the table

String createTableQuery = "CREATE TABLE IF NOT EXISTS Employee\_Records ("

+ "id INT PRIMARY KEY AUTO\_INCREMENT,"

+ "name VARCHAR(50),"

+ "designation VARCHAR(50),"

+ "salary DOUBLE)";

// Execute the query

statement.executeUpdate(createTableQuery);

System.out.println("Table created successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void insertRecords() {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"INSERT INTO Employee\_Records (name, designation, salary) VALUES (?, ?, ?)")) {

// Insert three records

insertRecord(preparedStatement, "John Doe", "Software Engineer", 75000);

insertRecord(preparedStatement, "Jane Smith", "Database Administrator", 80000);

insertRecord(preparedStatement, "Bob Johnson", "Project Manager", 90000);

System.out.println("Records inserted successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void insertRecord(PreparedStatement preparedStatement, String name, String designation, double salary)

throws SQLException {

preparedStatement.setString(1, name);

preparedStatement.setString(2, designation);

preparedStatement.setDouble(3, salary);

preparedStatement.executeUpdate();

}

private static void displayRecords() {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("SELECT \* FROM Employee\_Records")) {

System.out.println("Employee Records:");

// Display column headers

System.out.printf("%-5s%-20s%-20s%-10s%n", "ID", "Name", "Designation", "Salary");

System.out.println("------------------------------------------------------");

// Display records

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

String designation = resultSet.getString("designation");

double salary = resultSet.getDouble("salary");

System.out.printf("%-5d%-20s%-20s%-10.2f%n", id, name, designation, salary);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}

**24) Write a program to create Student Database. Write the following SQL queries**

**d) Create Table Student\_Records (Assume suitable data)**

**e) Insert 5 Records in the table**

**f) Display all Records g) Delete 2nd Record**

**h) Update 5th Record**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class StudentDatabase {

private static final String JDBC\_URL = "jdbc:mysql://localhost:3306/your\_database\_name";

private static final String USER = "your\_username";

private static final String PASSWORD = "your\_password";

public static void main(String[] args) {

createTable();

insertRecords();

displayRecords();

deleteRecord(2);

updateRecord(5, "Updated Name", "Updated Course", 85.5);

displayRecords();

}

private static void createTable() {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

Statement statement = connection.createStatement()) {

// SQL query to create the table

String createTableQuery = "CREATE TABLE IF NOT EXISTS Student\_Records ("

+ "id INT PRIMARY KEY AUTO\_INCREMENT,"

+ "name VARCHAR(50),"

+ "course VARCHAR(50),"

+ "marks DOUBLE)";

// Execute the query

statement.executeUpdate(createTableQuery);

System.out.println("Table created successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void insertRecords() {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"INSERT INTO Student\_Records (name, course, marks) VALUES (?, ?, ?)")) {

// Insert five records

insertRecord(preparedStatement, "Alice", "Mathematics", 90.0);

insertRecord(preparedStatement, "Bob", "Physics", 85.5);

insertRecord(preparedStatement, "Charlie", "Chemistry", 78.5);

insertRecord(preparedStatement, "David", "Biology", 92.5);

insertRecord(preparedStatement, "Eva", "Computer Science", 88.0);

System.out.println("Records inserted successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void insertRecord(PreparedStatement preparedStatement, String name, String course, double marks)

throws SQLException {

preparedStatement.setString(1, name);

preparedStatement.setString(2, course);

preparedStatement.setDouble(3, marks);

preparedStatement.executeUpdate();

}

private static void displayRecords() {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("SELECT \* FROM Student\_Records")) {

System.out.println("\nStudent Records:");

// Display column headers

System.out.printf("%-5s%-20s%-20s%-10s%n", "ID", "Name", "Course", "Marks");

System.out.println("------------------------------------------------------");

// Display records

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

String course = resultSet.getString("course");

double marks = resultSet.getDouble("marks");

System.out.printf("%-5d%-20s%-20s%-10.2f%n", id, name, course, marks);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void deleteRecord(int id) {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"DELETE FROM Student\_Records WHERE id = ?")) {

preparedStatement.setInt(1, id);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("\nRecord with ID " + id + " deleted successfully.");

} else {

System.out.println("\nNo record found with ID " + id);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void updateRecord(int id, String newName, String newCourse, double newMarks) {

try (Connection connection = DriverManager.getConnection(JDBC\_URL, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"UPDATE Student\_Records SET name = ?, course = ?, marks = ? WHERE id = ?")) {

preparedStatement.setString(1, newName);

preparedStatement.setString(2, newCourse);

preparedStatement.setDouble(3, newMarks);

preparedStatement.setInt(4, id);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

System.out.println("\nRecord with ID " + id + " updated successfully.");

} else {

System.out.println("\nNo record found with ID " + id);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}

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

import java.net.\*;

class inetdemo

{

Public static void main(string args[])

{

InetAddress a = InetAddress.getByName(“www.google.com”);

System.out.println(“Host Name=”+a.getHostName());

System.out.println(“Host Address=”+a.getHostAddress());

System.out.println(“Local Host =”+a.getLocalHost());

}

}

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

import java.net.\*;

class Urldemo

{

Public static void main(String args[])

{

URL a = new URL(“http://www.google.co,in:80/index.html”);

System.out.println(“Host =”+a.getHost());

System.out.println(“Protocol=”+a.getProtocol());

System.out.println(“Port No.=”+a.getPort());

System.out.println(“File Name=”+a.getFile);

System.out.println(“Query String =”+a.getQuery());

System.out.println(“Path =”+a.getPath());

}

}