

**Shram Sadhana Bombay Trust's
COLLEGE OF ENGINEERING AND TECHNOLOGY,**

BAMBHORI POST BOX NO. 94, JALGAON – 425001.(M.S.)

Included under section 2 (f) & 12 (B) of the UGC Act, 1956

NAAC Re-accredited Grade: A (3.14) (2nd Cycle)

ISO 9001: 2008 Certified

Phone No. (0257) 2258393, Fax No. (0257) 2258392

Website- www.sscoetjalgaon.ac.in

Email: sscoetjal@gmail.com



ISO 9001:2008

**DEPARTMENT OF COMPUTER APPLICATIONS
(M.C.A.)**

MCA

SSBT

COET

Laboratory Journal

Class: M.C.A-I, SEM-I, Academic Year: 2024-25

Subject: MCA-412: Lab on Java Programming

Academic Year: 2024 -25

University Exam. Seat No:

Name of the Faculty: Ms. Reeta V. Patil

Vision of the Institute

Today we carry the flame of quality education, knowledge and progressive technology for global societal development; tomorrow the flame will glow even brighter.

Mission of the Institute

To provide conducive environment for preparing competent, value added and patriotic engineers of integrity of par excellence to meet global standards for societal development.

DEPARTMENT OF COMPUTER APPLICATIONS (M.C.A.)

Vision of the Department

To develop skilled professionals with social values as per industry needs.

Mission of the Department

To provide a student-centered, conducive environment for preparing knowledgeable, skilled and value added professionals to stand out in a competitive world.

Department of Computer Applications (M.C.A.)

Program Outcomes (POs)

PO1: Computational Knowledge: Understand and apply mathematical foundation, computing and domain knowledge for the conceptualization of computing models from defined problems.

PO2: Problem Analysis: Ability to identify, critically analyze and formulate complex computing problems using fundamentals of computer science and application domains.

PO3: Design / Development of Solutions: Ability to transform complex business scenarios and contemporary issues into problems, investigate, understand and propose integrated solutions using emerging technologies.

PO4: Conduct Investigations of Complex Computing Problems: Ability to devise and conduct experiments, interpret data and provide well informed conclusions.

PO5: Modern Tool Usage: Ability to select modern computing tools, skills and techniques necessary for innovative software solutions.

PO6: Professional Ethics: Ability to apply and commit professional ethics and cyber regulations in a global economic environment.

PO7: Life-long Learning: Recognize the need for and develop the ability to engage in continuous learning as a Computing professional.

PO8: Project Management and Finance: Ability to understand, management and computing principles with computing knowledge to manage projects in multidisciplinary environments.

PO9: Communication Efficacy: Communicate effectively with the computing community as well as society by being able to comprehend effective documentations and presentations.

PO10: Societal & Environmental Concern: Ability to recognize economic, environmental, social, health, legal, ethical issues involved in the use of computer technology and other consequential responsibilities relevant to professional practice.

PO11: Individual and Team Work: Ability to work as a member or leader in diverse teams in multidisciplinary environment.

PO12: Innovation and Entrepreneurship: Identify opportunities, entrepreneurship vision and use of innovative ideas to create value and wealth for the betterment of the individual and society.

Department of Computer Applications (M.C.A.)

Program Educational Objectives (PEOs)

PEO 1: Core Knowledge -The computing professional graduate will have the knowledge of basic science and Engineering skills, Humanities, social science, management and conceptual and practical understanding of core computer applications area with project development.

PEO 2: Employment/ Continuing Education – The computing professional graduate will have the knowledge of Industry-based technical skills to succeed in entry level professional position at various industries as well as in academics.

PEO 3: Professional Competency - The computing professional graduate will have the ability to communicate effectively in English, to accumulate and disseminate the knowledge and to work effectively in a team with a sense of social awareness.

Department of Computer Applications (M.C.A.)

Program Specific Outcomes (PSOs)

PSO 1: The computing professional graduate will be able to apply knowledge of computer science in practice to identify, critically analyze, formulate and develop computer applications using modern computing tools and techniques and will use these tools with dexterity.

PSO 2: The computing professional graduate will be able to design computing systems to meet desired needs within realistic constraints such as safety, security and applicability. These systems will function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with positive attitude.

PSO 3: The computing professional graduate will be able to appreciate the importance of goal setting and recognize the need for life-long learning with good communication skills.

Department of Computer Applications (M.C.A.)

MCA-412: Lab on Java Programming

M.C.A-I SEM-I

Course Outcomes

CO1: Write Java programs using inner classes and static fields in the implementation of Java applications.

CO2: Develop Java applications for GUI development and event handling.

CO3: Develop database applications using JDBC

CO4: Apply Java programming constructs to develop simple programs that solve basic computational problems.

CO-PO and PSO Mappings

Course Outcomes	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O 1	PS O2	PS O3
CO1:	3	3			3							3	3	2	
CO2:	3	3	3	3	3				2		3	3	3	3	
CO3:	3	3	3	3	3				2		3	3	3	3	
CO4:	3	3			3		2					3	3	3	

Department of Computer Applications (M.C.A.)

MCA-412: Lab on Java Programming

List of Practicals

Sr.No.	Name of the Practical	Date of Performance	Date of Completion	Sign of Teacher
1	Implement program that demonstrates program structure of java with use of arithmetical and 21 Logical implementation			
2.	Implement a program that demonstrate string operations using String and String Buffer class			
3	Implement a program that demonstrate inner class and static fields			
4.	Implement a program that demonstrate inheritance, polymorphism.			
5.	Implement program that demonstrate 2D shapes on frames			
6	Implement a program that demonstrate color and Fonts..			
7	Implement a program to illustrate use of various Swing components.			
8	Implement a program that demonstrate use of dialog box and menus			
9	Implement a program that demonstrate event Handling for various types of events.			
10	Implement a program to illustrate multithreading.			
11	Implement a program to illustrate exception handling			
12	Implement a program to demonstrate use of File class			
13	Implement a program that demonstrate JDBC on application			
14	Implement a program that demonstrate packagecreation and use in program.			

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 01

DOP:

DOC:

Title: Implement a program that demonstrates program structure of java with use of arithmetical and 21 logical implementation.

Code:-

```
public class Assignment1
{
    public static void main(String[] args)
    {
        // initializing variables
        int num1 = 20, num2 = 10, sum = 0, diff = 0, multi=0;
        float div=0;
        System.out.println("num1 = " + num1);
        System.out.println("num2 = " + num2);
        sum = num1 + num2;
        System.out.println("The sum = " + sum);
        diff = num1 - num2;
        System.out.println("The diff = " + diff);
        multi = num1 * num2;
        System.out.println("The multi = " + multi);
        div = num1 / num2;
        System.out.println("The div = " + div);
        if ((num1==20) && (num2==10))// You can also use || operator
        {
            System.out.println("Both True");
        }
        else
            System.out.println("Both Not True");
    }
}
```

OUTPUT:-

```
Console ×
<terminated> Assignment1 [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (Dec 3, 2024, 10:05:40 PM – 10:05:41 PM) [pid: 11632]
num1 = 20
num2 = 10
The sum = 30
The diff = 10
The multi = 200
The div = 2.0
Both True
```

Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 02

DOP:

DOC:

Title: Implement a program that demonstrates string operations using String andString Buffer class.

CODE:-

```
package assignment2;
import java.io.*;
public class Assignment2
{
    public static void main(String[] args)
    {
        try
        {
            DataInputStream d= new DataInputStream(System.in);
            System.out.println("\n enter the 1st String ");
            String s=d.readLine();
            //String Functions
            int y=s.length();
            System.out.println("\n length of string is "+y);
            String z=s.toUpperCase();
            System.out.println("\n string in upper case "+z);
            String l=s.toLowerCase();
            System.out.println("\n string in lower case "+l);
            char m=s.charAt(3);
            System.out.println("\n char at 3rd index is "+m);
            String o=s.replace('a','b');
            System.out.println("\n replaced string is "+o);
            String n=s.substring(2,5);
            System.out.println("\n sub string from 2 to 5 index is "+n);
            System.out.println("\n enter the character to find index");
            String s2=d.readLine();
            int a=s.indexOf(s2);
            System.out.println("\n index of char is "+a);
            System.out.println("\n enter the character to find last index");
            String s3=d.readLine();
            int b=s.lastIndexOf(s3);
            System.out.println("\n last index of char is "+b);
            System.out.println("\n enter the 2nd String ");
            String s1=d.readLine();

            String p=s.concat(s1);

            System.out.println("\n concated string is "+p);
```

```

boolean b1=s.equals(s1);
if(b1==true)
{
    System.out.println("\n strings are equal ");
}
else
{
    System.out.println("\n strings are  not equal ");
}
//StringBuffer Functions
StringBuffer sf = new StringBuffer("Coding Atharva");
System.out.println("\n String = "+sf); // Will Print the string
System.out.println("\n Length = "+sf.length() ); // total numbers of characters
System.out.println("\n Length = "+sf.capacity() ); // total allocated capacity
sf.setLength(6); // Sets the length and destroy the remaining characters
System.out.println("\n After setting length String = "+sf);
sf.setCharAt(0,'K'); // It will change character at specified position
System.out.println("\n SetCharAt String = "+sf);
sf.setCharAt(0,'C');
int a1 = 7;
sf.append(a1); // It concatenates the other data type value
System.out.println("\n Appended String = "+sf);
sf.insert(6," Atharva"); // used to insert one string or char or object
System.out.println("\n Inserted String = "+sf);
sf.reverse();
System.out.println("\n Reverse String = "+sf);
}
catch(Exception e)
{
    System.out.println(""+e);
}
}
}

```

OUTPUT:-

```

enter the 1st String
manojkumar
length of string is 10
string in upper case MANOJKUMAR
string in lower case manojkumar
char at 3rd index is o
replaced string is mbnojkumbr
sub string from 2 to 5 index is noj
enter the character to find index
a
index of char is 1
enter the character to find last index

```

a
last index of char is 8
enter the 2nd String
sonawane
concatated string is manojkumarsonawane
strings are not equal
String = Coding Atharva
Length = 14
Length = 30
After setting length String = Coding
SetCharAt String = Koding
Appended String = Coding7
Inserted String = Coding Atharva7
Reverse String = 7avrahtA gnidoC

Submitted By :

Sign :

Name :

Roll No :

Checked By :

<Name of Faculty>

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 03

DOP:

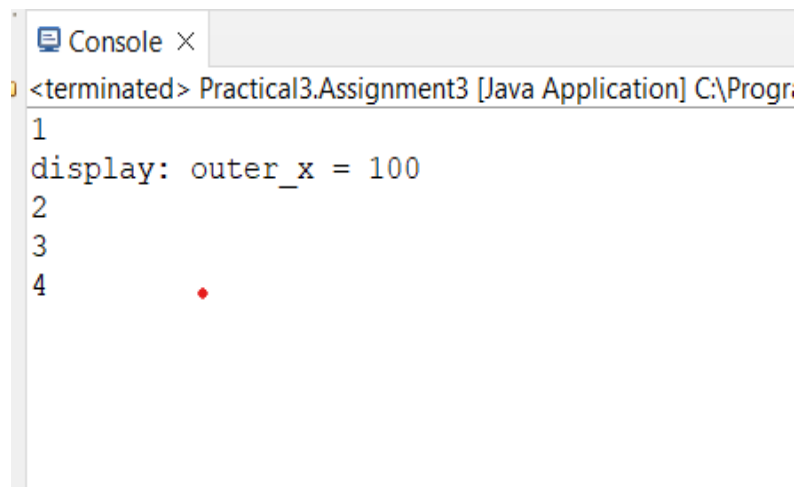
DOC:

Title: Implement a program that demonstrate inner class and static fields.

CODE:-

```
package assignment3;
class Outer
{
    int outer_x = 100;
    void test()
    {
        Inner inner = new Inner(); inner.display();
    }
    static int count=0;//will get memory only once and retain its value
    Outer()
    {
        count++;//incrementing the value of static variable
        System.out.println(count);
    }
    class Inner
    {
        void display()
        {
            System.out.println("display: outer_x = " + outer_x);
        }
    }
}
public class Assignment3
{
    public static void main(String[] args)
    {
        Outer outer = new Outer();
        outer.test();
        //creating objects
        Outer o1=new Outer();
        Outer o2=new Outer();
        Outer o3=new Outer();
    }
}
```

OUTPUT:-



```
<terminated> Practical3.Assignment3 [Java Application] C:\Progr.  
1  
display: outer_x = 100  
2  
3  
4
```

Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 04

DOP:

DOC:

Title: Implement a program that demonstrate inheritance, polymorphism.

CODE:-

```
package mca.firstyear;

public class assignment4 {

    static class Animal {
        public void move() {
            System.out.println("Animals can move");
        }
    }

    static class Dog extends Animal {
        // Method Overriding
        public void move() {
            System.out.println("Dogs can walk and run");
        }

        // Method Overloading
        void add(int a, int b) {
            int s = a + b;
            System.out.println("Sum=" + s);
        }

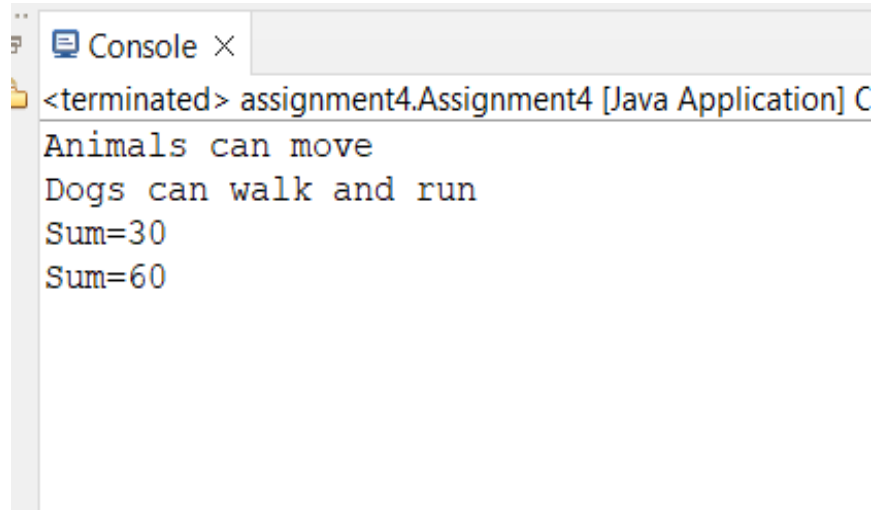
        void add(int a, int b, int c) {
            int s = a + b + c;
            System.out.println("Sum=" + s);
        }
    }

    public static class Assignment4 {
        public static void main(String[] args) {
            Animal a = new Animal();
            Animal b = new Dog();
            a.move();
            b.move();

            Dog d = new Dog();
            d.add(10, 20);
        }
    }
}
```

```
d.add(10, 20, 30);  
  
    }  
}  
}
```

OUTPUT:-



The screenshot shows a Java IDE console window titled "Console x". The output text is as follows:

```
<terminated> assignment4.Assignment4 [Java Application] C  
Animals can move  
Dogs can walk and run  
Sum=30  
Sum=60
```

Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 05

DOP:

DOC:

Title: Implement a program that demonstrate 2D shapes on frames

CODE :-

```
package com.example;

import java.awt.Color;

import java.awt.Graphics;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class ShapeDemo extends JPanel {

    // Override the paintComponent method to draw 2D shapes

    @Override    protected void paintComponent(Graphics g) {

        super.paintComponent(g); // Call the superclass method to ensure proper rendering

        // Draw a rectangle

        g.setColor(Color.BLUE);

        g.fillRect(50, 50, 200, 100); // x, y, width, height

        // Draw an oval (circle in this case)

        g.setColor(Color.RED);

        g.fillOval(300, 50, 100, 100); // x, y, width, height

        // Draw a line

        g.setColor(Color.GREEN);

        g.drawLine(50, 200, 350, 200); // x1, y1, x2, y2
```



```
// Draw a triangle (polygon)

int[] xPoints = {200, 250, 300};

int[] yPoints = {250, 150, 250};

g.setColor(Color.MAGENTA);

g.fillPolygon(xPoints, yPoints, 3); // x points, y points, number of points
}

// Main method to create the frame and display the panel

public static void main(String[] args) {

    // Create a frame to hold the panel

    JFrame frame = new JFrame("2D Shapes Demo");

    // Set up the frame

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    frame.setSize(500, 400); // Width and height of the window

    // Create an instance of the ShapeDemo panel and add it to the frame

    ShapeDemo panel = new ShapeDemo();

    frame.add(panel);

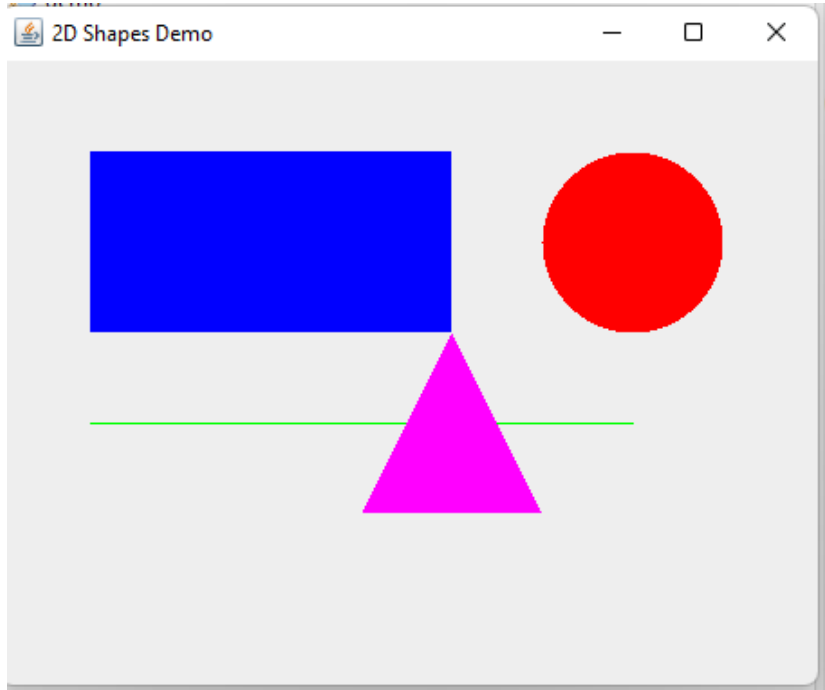
    // Make the frame visible

    frame.setVisible(true);

}

}
```

OUTPUT-



Submitted By :

Sign :

Name :

Roll No :

Checked By :

<Name of Faculty>

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 06

DOP:

DOC:

Title: Implement a program that demonstrates color and fonts.

CODE:-

```
package pro;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class ColorAndFontDemo extends JPanel {

    // Override the paintComponent method to draw with colors and fonts

    @Override

    protected void paintComponent(Graphics g) {

        super.paintComponent(g); // Call the superclass method to ensure proper rendering

        // Set background color

        setBackground(Color.WHITE);

        // Draw a colored rectangle with different border and fill color

        g.setColor(Color.CYAN); // Set fill color

        g.fillRect(50, 50, 200, 100); // x, y, width, height

        g.setColor(Color.DARK_GRAY); // Set border color

        g.drawRect(50, 50, 200, 100); // x, y, width, height (border)

        // Draw text with different font styles and sizes
```

```
g.setColor(Color.RED); // Set color for text

g.setFont(new Font("Serif", Font.BOLD, 20)); // Set font (Serif, Bold, Size 20)

g.drawString("This is a BOLD Serif Font", 50, 200);

g.setColor(Color.BLUE); // Set color for next text

g.setFont(new Font("SansSerif", Font.ITALIC, 24)); // Set font (SansSerif, Italic, Size
24)

g.drawString("This is an ITALIC SansSerif Font", 50, 250);


g.setColor(Color.GREEN); // Set color for next text

g.setFont(new Font("Monospaced", Font.PLAIN, 18)); // Set font (Monospaced, Plain,
Size 18)

g.drawString("This is a PLANE Monospaced Font", 50, 300);


// Draw a string with custom font and size

g.setColor(Color.MAGENTA);

g.setFont(new Font("Arial", Font.PLAIN, 30)); // Set font (Arial, Plain, Size 30)

g.drawString("Magenta Text in Arial", 50, 350);

}

// Main method to create the frame and display the panel

public static void main(String[] args) {

    // Create a frame to hold the panel

    JFrame frame = new JFrame("Color and Font Demo");

    // Set up the frame

    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    frame.setSize(600, 500); // Width and height of the window

    // Create an instance of the ColorAndFontDemo panel and add it to the frame
```

```
ColorAndFontDemo panel = new ColorAndFontDemo();  
  
frame.add(panel);  
  
// Make the frame visible  
  
frame.setVisible(true);  
  
}  
  
}
```

OUTPUT-



Submitted By :

Sign :

Name :

Roll No :

Checked By :

<Name of Faculty>

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 07

DOP:

DOC:

Title: Implement a program to illustrate use of various swing components.

CODE:-

```
package project;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SwingComponentsExample {

    public static void main(String[] args) {
        // Create the main JFrame
        JFrame frame = new JFrame("Swing Components Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 400);
        frame.setLayout(new FlowLayout());

        // Add JLabel and JTextField for Name
        JLabel nameLabel = new JLabel("Enter your name:");
        JTextField nameField = new JTextField(20);
        frame.add(nameLabel);
        frame.add(nameField);

        // Add JCheckBox
        JCheckBox subscribeCheckBox = new JCheckBox("Subscribe to Newsletter");
        frame.add(subscribeCheckBox);

        // Add JRadioButtons for Gender
        JLabel genderLabel = new JLabel("Select your gender:");
        JRadioButton maleButton = new JRadioButton("Male");
        JRadioButton femaleButton = new JRadioButton("Female");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleButton);
        genderGroup.add(femaleButton);
        frame.add(genderLabel);
        frame.add(maleButton);
        frame.add(femaleButton);
    }
}
```

```

// Add JComboBox for Country Selection
JLabel countryLabel = new JLabel("Select your country:");

String[] countries = {"India", "USA", "UK", "Canada"};
JComboBox<String> countryComboBox = new JComboBox<>(countries);
frame.add(countryLabel);
frame.add(countryComboBox);

// Add JTextArea to display output
JTextArea outputArea = new JTextArea(5, 30);
outputArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(outputArea);
frame.add(scrollPane);

// Add JButton
JButton submitButton = new JButton("Submit");
frame.add(submitButton);

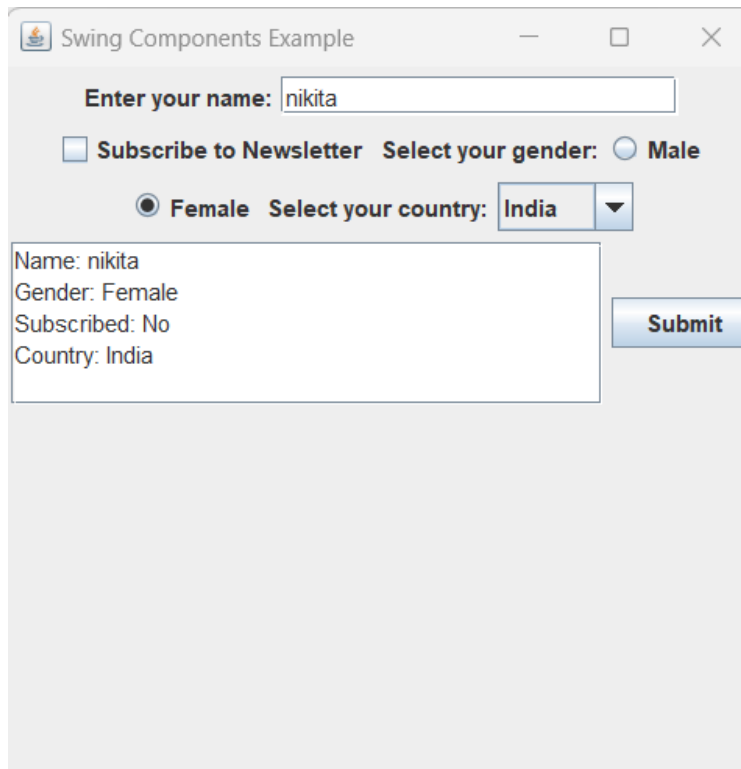
// Add ActionListener for Submit Button
submitButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String gender = maleButton.isSelected()?"Male":(femaleButton.isSelected()
            ? "Female" : "Not Selected");
        boolean isSubscribed = subscribeCheckBox.isSelected();
        String country = (String) countryComboBox.getSelectedItem();

        // Display the collected input in the JTextArea
        outputArea.setText("Name: " + name + "\n"
            + "Gender: " + gender + "\n"
            + "Subscribed: " + (isSubscribed ? "Yes" : "No") + "\n"
            + "Country: " + country);
    }
});

// Set frame visibility
frame.setVisible(true);
}
}

```

OUTPUT-



The screenshot shows a Java Swing window titled "Swing Components Example". Inside the window, there is a form with the following elements:

- A text field labeled "Enter your name:" containing the text "nikita".
- A checkbox labeled "Subscribe to Newsletter" which is unchecked.
- Radio buttons for "Select your gender:" with "Male" and "Female" options. "Female" is selected.
- A dropdown menu for "Select your country:" with "India" selected.
- A "Submit" button.
- A text area displaying the form data: "Name: nikita", "Gender: Female", "Subscribed: No", and "Country: India".

Submitted By :

Sign :

Name :

Roll No :

Checked By :

<Name of Faculty>

Ms. Reeta V. Patil.


```

        JOptionPane.showMessageDialog(frame,
            "You selected: " + selectedFile.getAbsolutePath(),
            "File Opened",
            JOptionPane.INFORMATION_MESSAGE);
    } else if (result == JFileChooser.CANCEL_OPTION) {
        // Handle cancel action
        JOptionPane.showMessageDialog(frame,
            "Open operation was canceled.",
            "Action Canceled",
            JOptionPane.WARNING_MESSAGE);
    }
}

});
// Add action listener for "Exit" menu item
exitItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Exit the application
        System.exit(0);
    }
});
// Add menu items to the "File" menu
fileMenu.add(openFileItem);
fileMenu.addSeparator(); // Add a separator
fileMenu.add(exitItem);
// Add the "File" menu to the menu bar
menuBar.add(fileMenu);
// Create the "Options" menu
JMenu optionsMenu = new JMenu("Options");

// Create menu items for the "Options" menu
JMenuItem showMessageItem = new JMenuItem("Show Message");
// Add action listener for "Show Message" menu item
showMessageItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // Show a dialog box with a message
        JOptionPane.showMessageDialog(frame,
            "Hello! This is a dialog box.",
            "Message",
            JOptionPane.INFORMATION_MESSAGE);
    }
});

```

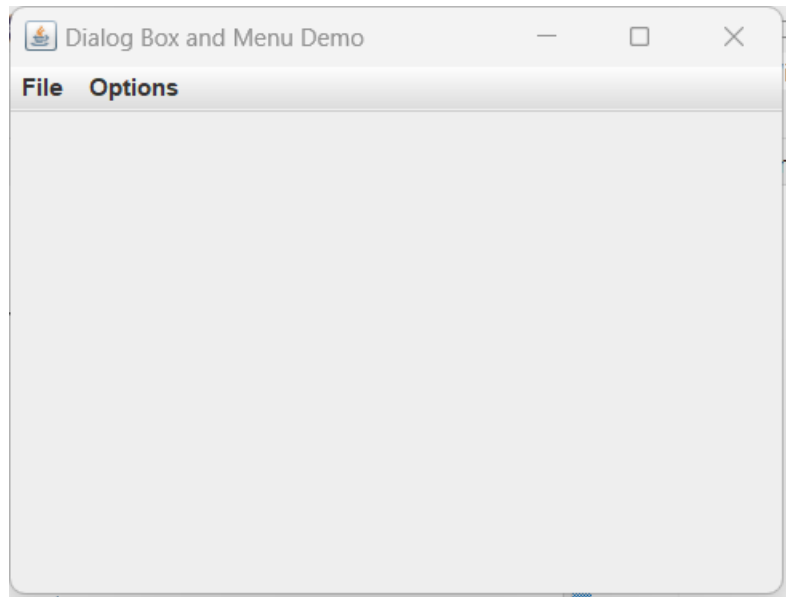
```
    }

});
// Add menu items to the "Options" menu
optionsMenu.add(showMessageItem);

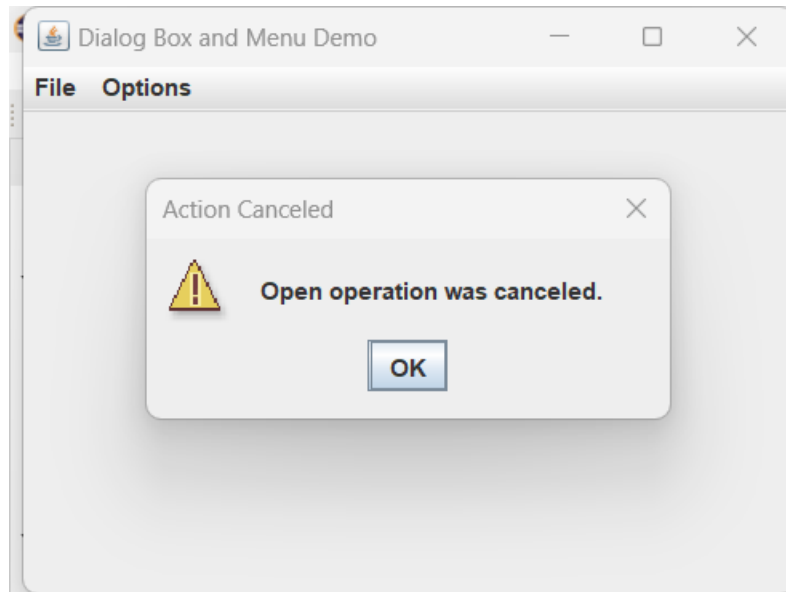
// Add the menus to the menu bar
menuBar.add(optionsMenu);
// Set the menu bar to the frame
frame.setJMenuBar(menuBar);
// Set frame visibility
frame.setVisible(true);
}
}
```

OUTPUT-

Menu



Dialog box



Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 09

DOP:

DOC:

Title: Implement a program that demonstrate event handling for various types of events.

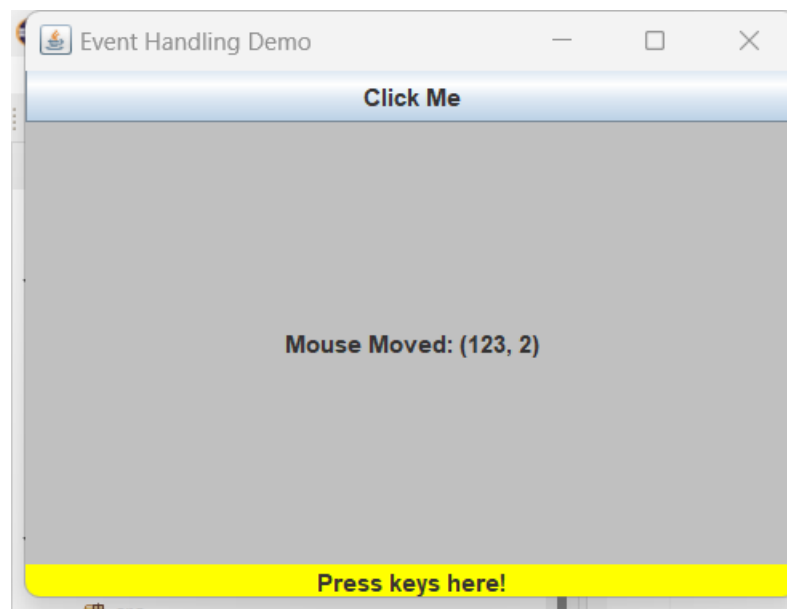
CODE:-

```
package demo;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
@SuppressWarnings("serial")
public class EventHandlingDemo extends JFrame {
    private JButton button;
    private JLabel mouseLabel, keyLabel;
    public EventHandlingDemo() {
        // Set up the frame
        setTitle("Event Handling Demo");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        // Create a button and add an ActionListener
        button = new JButton("Click Me");
        button.addActionListener(e -> JOptionPane.showMessageDialog(this, "Button Clicked!"));
        add(button, BorderLayout.NORTH);
        // Create a label for mouse events
        mouseLabel = new JLabel("Move the mouse here!", SwingConstants.CENTER);
        mouseLabel.setOpaque(true);
        mouseLabel.setBackground(Color.LIGHT_GRAY);
        mouseLabel.addMouseListener(new MouseMotionAdapter() {
            @Override
            public void mouseMoved(MouseEvent e) {
                mouseLabel.setText("Mouse Moved: (" + e.getX() + ", " + e.getY() + ")");
            }
        });
        add(mouseLabel, BorderLayout.CENTER);
        // Create a label for key events
        keyLabel = new JLabel("Press keys here!", SwingConstants.CENTER);
        keyLabel.setFocusable(true);
        keyLabel.setOpaque(true);
        keyLabel.setBackground (Color.YELLOW);
    }
}
```

```
keyLabel.addKeyListener(new KeyAdapter() {

    @Override
    public void keyPressed(KeyEvent e) {
        keyLabel.setText("Key Pressed: " + KeyEvent.getKeyText(e.getKeyCode()));
    }
    @Override
    public void keyReleased(KeyEvent e) {
        keyLabel.setText("Key Released: " + KeyEvent.getKeyText(e.getKeyCode()));
    }
});
add(keyLabel, BorderLayout.SOUTH);
// Display the frame
setVisible(true);
}
public static void main(String[] args) {
    SwingUtilities.invokeLater(EventHandlingDemo::new);
}
```

OUTPUT:-



Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 10

DOP:

DOC:

Title: Implement a program to illustrate multithreading.

CODE:-

```
package demo6;
// Thread implementation by extending the Thread class
class Thread1 extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread extending Thread class: " + i);
            try {
                Thread.sleep(500); // Pause for 500 milliseconds
            }
            catch (InterruptedException e) {
                System.out.println("Thread interrupted: " + e.getMessage());
            }
        }
    }
}

// Thread implementation by implementing the Runnable interface
class MyRunnable implements Runnable {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println("Thread implementing Runnable: " + i);
            try {
                Thread.sleep(500); // Pause for 500 milliseconds
            }
            catch (InterruptedException e) {
                System.out.println("Thread interrupted: " + e.getMessage());
            }
        }
    }
}

public class MyThread {
    public static void main(String[] args) {
        // Create a thread by extending Thread
        MyThread thread1 = new MyThread()
        // Create a thread by implementing Runnable
        Thread thread2 = new Thread(new MyRunnable());
        // Start the threads
    }
}
```

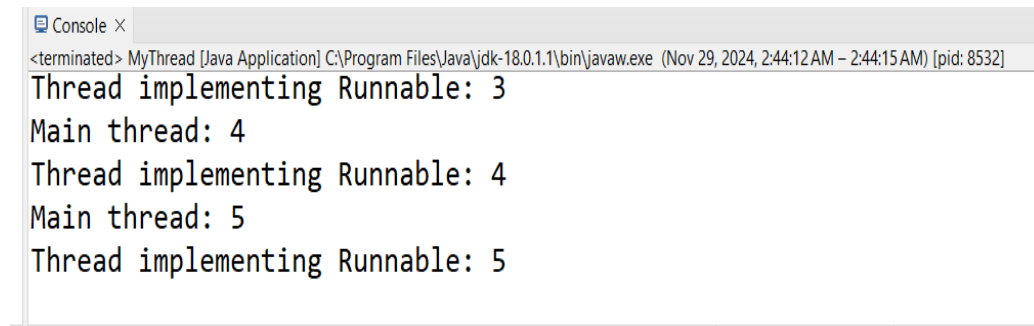
```

thread1.start();
thread2.start();

// Main thread work
for (int i = 1; i <= 5; i++) {
    System.out.println("Main thread: " + i);
    try {
        Thread.sleep(500); // Pause for 500 milliseconds
    }
    catch (InterruptedException e) {
        System.out.println("Main thread interrupted: " + e.getMessage());
    }
}
private void start() {
    // TODO Auto-generated method stub
}
}

```

OUTPUT:-



```

Console x
<terminated> MyThread [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (Nov 29, 2024, 2:44:12 AM - 2:44:15 AM) [pid: 8532]
Thread implementing Runnable: 3
Main thread: 4
Thread implementing Runnable: 4
Main thread: 5
Thread implementing Runnable: 5

```

Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 11

DOP:

DOC:

Title: Implement a program to illustrate exception handling.

CODE:-

```
package demo6;
public class Assignment11
{
    public static void main(String[] args)
    {
        try
        {
            @SuppressWarnings("unused")
            int i=2/0;
            int a[]=new int[5];
            a[10]=30;
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("ArrayIndexOutOfBoundsException");
        }
        catch(ArithmeticException e)
        {
            System.out.println("ArithmeticException");
        }
        catch(Exception e)
        {
            System.out.println("Exception");
        }
        finally
        {
            System.out.println("Finally");
        }
    }
}
```

OUTPUT:-



The screenshot shows a Java console window titled "Console X". The status bar at the top indicates the application is terminated: "<terminated> Assignment11 [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (Nov 29, 2024, 5:23:10 AM - 5:23:10 AM) [pid: 13428]". The main content area displays the text "ArithmeticException" in blue, underlined font, followed by "Finally" in black font on the next line.

Submitted By :

Sign :

Name :

Roll No :

Checked By :

<Name of Faculty>

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 12

DOP:

DOC:

Title: Implement a program to demonstrate use of File class.

CODE:-

```
package demo6;
import java.io.*;
public class Assignment12 {
    public static void main(String[] args) {
        FileInputStream sourceStream = null;
        FileOutputStream targetStream = null;

        try {
            // Ensure the source file exists and add content
            File sourceFile = new File("sourcefile.txt");
            if (!sourceFile.exists()) {
                sourceFile.createNewFile();
                try (FileWriter writer = new FileWriter(sourceFile)) {
                    writer.write("This is sample content for the source file.");
                    System.out.println("Content written to source file.");
                }
            }

            // Ensure the target file exists
            File targetFile = new File("targetfile.txt");
            if (!targetFile.exists()) {
                targetFile.createNewFile();
                System.out.println("Target file created: " + targetFile.getAbsolutePath());
            }

            // Open streams for reading and writing
            sourceStream = new FileInputStream(sourceFile);
            targetStream = new FileOutputStream(targetFile);
```

```

        // Reading source file and writing content to target file byte by byte
        int temp;
        while ((temp = sourceStream.read()) != -1) {
            targetStream.write(temp);
        }
        System.out.println("Content copied successfully from source to target.");

    } catch (IOException e) {
        System.out.println("An IOException occurred: " + e.getMessage());
    } finally {
        try {
            if (sourceStream != null) sourceStream.close();
            if (targetStream != null) targetStream.close();
        } catch (IOException e) {
            System.out.println("Error closing streams: " + e.getMessage());
        }
    }

    // File class operations and displaying data
    try {
        // Display data of the source file
        System.out.println("\n--- Source File Data ---");
        displayFileContent("sourcefile.txt");

        // Display data of the target file
        System.out.println("\n--- Target File Data ---");
        displayFileContent("targetfile.txt");
    } catch (IOException e) {
        System.out.println("Error reading files: " + e.getMessage());
    }

    // File class operations
    File f = new File("sourcefile.txt");
    System.out.println("\n--- File Properties ---");
    System.out.println("The name of the file is: " + f.getName());
    System.out.println("The absolute path of the file is: " + f.getAbsolutePath());

```

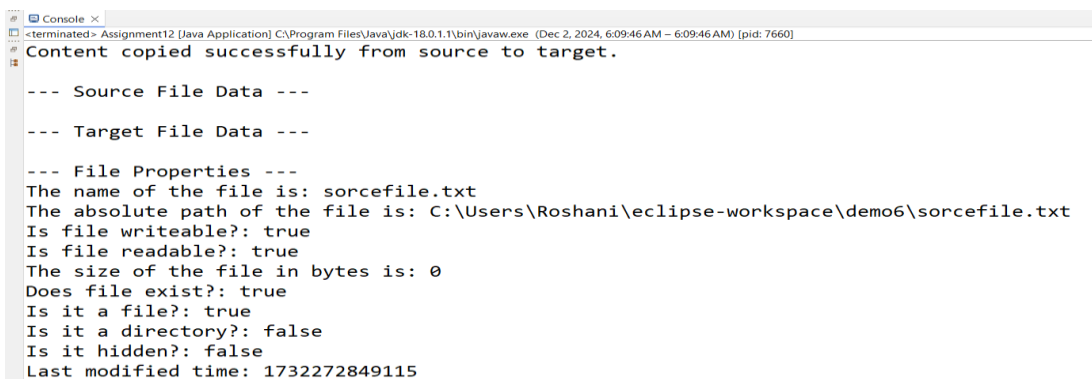
```

        System.out.println("Is file writeable?: " + f.canWrite());
        System.out.println("Is file readable?: " + f.canRead());
        System.out.println("The size of the file in bytes is: " + f.length());
        System.out.println("Does file exist?: " + f.exists());
        System.out.println("Is it a file?: " + f.isFile());
        System.out.println("Is it a directory?: " + f.isDirectory());
        System.out.println("Is it hidden?: " + f.isHidden());
        System.out.println("Last modified time: " + f.lastModified());
    }

    // Helper method to display file content
    private static void displayFileContent(String fileName) throws IOException {
        try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        }
    }
}

```

OUTPUT:-



```

Content copied successfully from source to target.

--- Source File Data ---

--- Target File Data ---

--- File Properties ---
The name of the file is: sorcefile.txt
The absolute path of the file is: C:\Users\Roshani\eclipse-workspace\demo6\sorcefile.txt
Is file writeable?: true
Is file readable?: true
The size of the file in bytes is: 0
Does file exist?: true
Is it a file?: true
Is it a directory?: false
Is it hidden?: false
Last modified time: 1732272849115

```

Submitted By :

Checked By :

Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.

SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 13

DOP:

DOC:

Title: Implement a program that demonstrates JDBC on application.

Step 2: Prerequisites

- **Database Setup:** Ensure you have a database installed (e.g., MySQL) and running. Create a database (testdb) in your database server.
- **JDBC Driver:** Download the JDBC driver for your database (e.g., MySQL mysql-connector-java-X.X.X.jar).

Step 2: Create a New Project in Eclipse

1. Open Eclipse.
2. Go to **File > New > Java Project**.
3. Enter the project name, e.g., JDBCdemoApp, and click **Finish**.

Step 3: Add the JDBC Driver to Your Project

1. Right-click on the project in the **Project Explorer**.
2. Select **Build Path > Configure Build Path**.
3. Go to the **Libraries** tab and click **Add External JARs**.
4. Select the JDBC driver (mysql-connector-java-X.X.X.jar) and click **Open**.
5. Click **Apply and Close**.

Step 4: Write the JDBC Code

1. In the **Project Explorer**, right-click on the src folder.
2. Select **New > Class**.
3. Enter the class name, e.g., JDBCdemo, and click **Finish**.
4. Write the following code in the JDBCdemo.java file:

Step 5: Run the Program

1. Save the file (Ctrl + S).
2. Right-click on the class file in the **Project Explorer**.
3. Select **Run As > Java Application**.

Notes:

1. **Database URL:** Replace `localhost:3306/testdb` with your database server and database name.
2. **Credentials:** Use your database username and password.
3. **Dependencies:** Ensure the JDBC driver matches your database version.

CODE:-

```
import java.sql.*;

public class JDBCdemo {

    public static void main(String[] args) {

        // Database configuration

        String url = "jdbc:mysql://localhost:3306/testdb"; // Replace 'testdb' with your database
        name

        String user = "root"; // Replace with your MySQL username

        String password = "password"; // Replace with your MySQL password


        // SQL Statements

        String createTableSQL = "CREATE TABLE IF NOT EXISTS employees (" +

            "id INT AUTO_INCREMENT PRIMARY KEY," +

            "name VARCHAR(50) NOT NULL," +

            "position VARCHAR(50) NOT NULL," +

            "salary DOUBLE NOT NULL)";

        String insertSQL = "INSERT INTO employees (name, position, salary) VALUES (?, ?,
        ?)";

        String selectSQL = "SELECT * FROM employees";

        try (Connection connection = DriverManager.getConnection(url, user, password)) {

            System.out.println("Connected to the database!");
```

```
// Create table

try (Statement statement = connection.createStatement()) {

    statement.execute(createTableSQL);

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

}

// Insert data

try (PreparedStatement preparedStatement = connection.prepareStatement(insertSQL))
{

    preparedStatement.setString(1, "John Doe");

    preparedStatement.setString(2, "Software Engineer");

    preparedStatement.setDouble(3, 75000.00);

    preparedStatement.executeUpdate();

    preparedStatement.setString(1, "Jane Smith");

    preparedStatement.setString(2, "Manager");

    preparedStatement.setDouble(3, 90000.00);

    preparedStatement.executeUpdate();

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

}

// Retrieve and display data

try (Statement statement = connection.createStatement();

    ResultSet resultSet = statement.executeQuery(selectSQL)) {

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

    while (resultSet.next()) {

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

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



```

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

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


        System.out.printf("ID: %d, Name: %s, Position: %s, Salary: %.2f%n", id, name,
            position, salary);

    }

}

} catch (SQLException e) {

    e.printStackTrace();

}

}

}

```

OUTPUT:-

Connected to the database!
Table 'employees' created successfully.
Data inserted successfully.
Employee Data:
ID: 1, Name: John Doe, Position: Software Engineer, Salary: 75000.00
ID: 2, Name: Jane Smith, Position: Manager, Salary: 90000.

Submitted By :

Checked By :

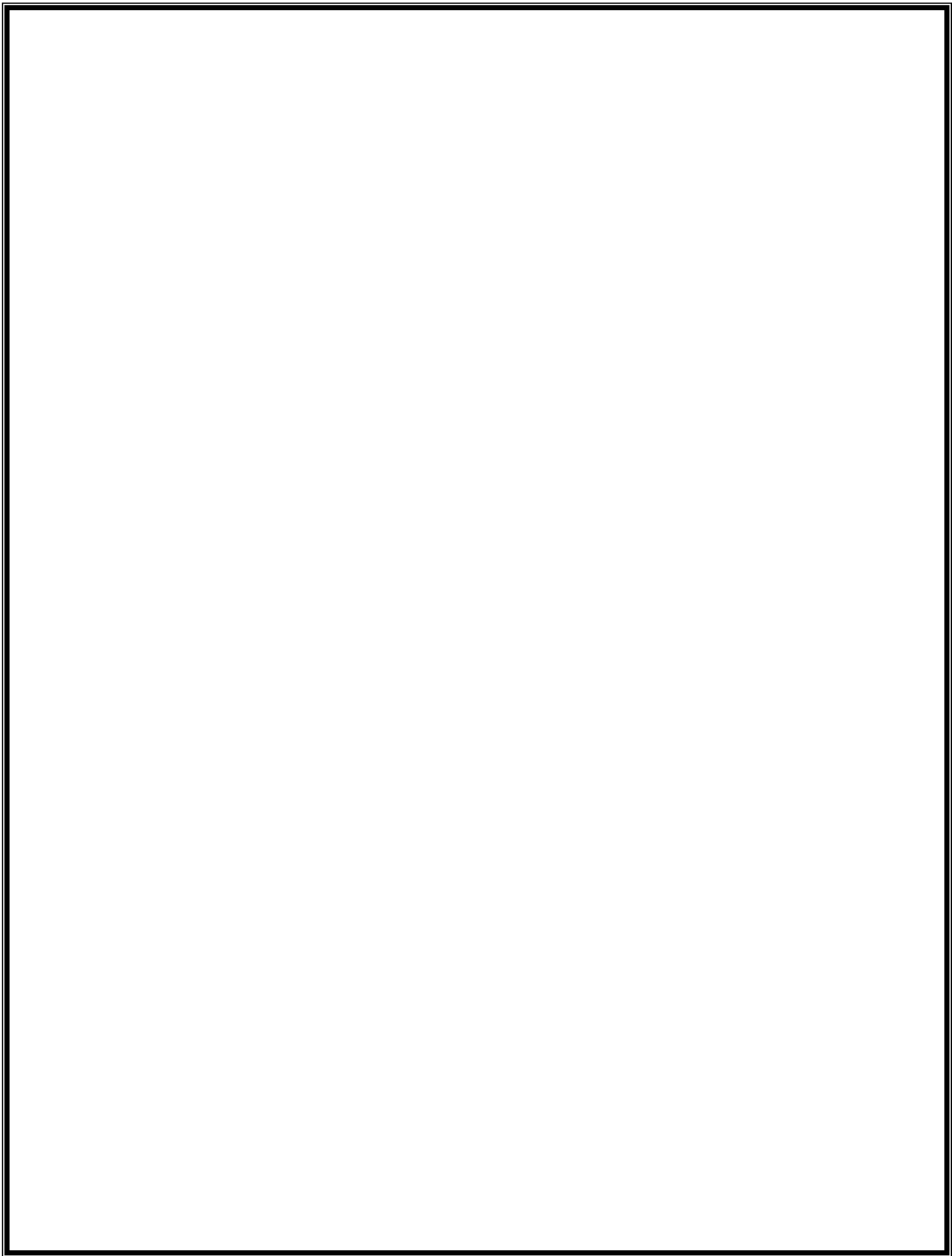
Sign :

Name :

<Name of Faculty>

Roll No :

Ms. Reeta V. Patil.



SSBT's College of Engineering & Technology, Bambhori, Jalgaon
Department of Computer Applications

Practical: 14

DOP:

DOC:

Title: Implement a program that demonstrate package creation and use in program.

File Structure:

```

  ▾ 📁 src
    ▾ 📁 mainpackage
      > 📄 MainProgram.java
    ▾ 📁 mypackage
      > 📄 MyClass.java
```

CODE:-

File Name : MainPackage.java

```
package mainpackage;

import mypackage.MyClass; // Import the class from mypackage

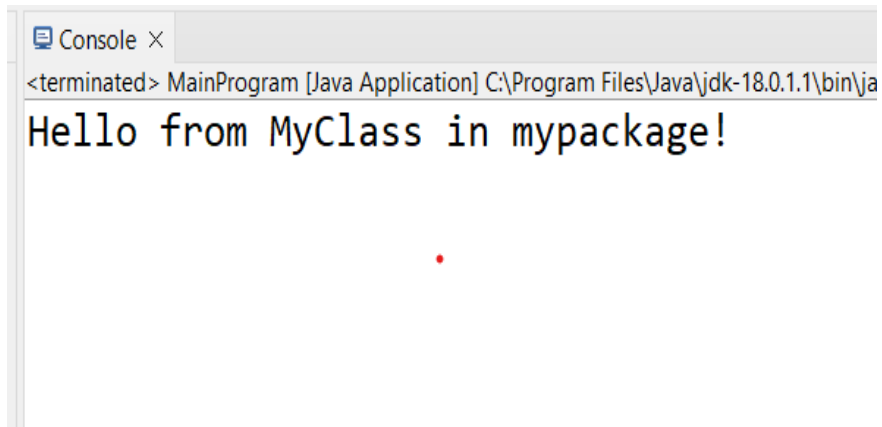
public class MainProgram {
    public static void main(String[] args) {
        MyClass myClass = new MyClass(); // Create an instance of MyClass
        myClass.displayMessage();      // Call the method
    }
}
```

File Name : MyClass.java

```
package mypackage;

public class MyClass {
    public void displayMessage() {
        System.out.println("Hello from MyClass in mypackage!");
    }
}
```

OUTPUT:-

A screenshot of a Java console window. The title bar reads "Console X". The text inside the console shows the program has terminated and displays the output: "Hello from MyClass in mypackage!".

```
<terminated> MainProgram [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\ja
Hello from MyClass in mypackage!
```

Submitted By :

Sign :

Name :

Roll No :

Checked By :

<Name of Faculty>

Ms. Reeta V. Patil.

