

Slip no 16:

1. o/p→

```
import java.util.*;
import java.io.*;

public class Collection {

    public static void main(String args[]) throws IOException {
        Set ts = new TreeSet();
        ts.add("Red");
        ts.add("Blue");
        ts.add("Yellow");
        ts.add("Pink");
        ts.add("Baby Pink");
        System.out.println("TreeSet in ascending order: " + ts);
    }
}
```

2. o/p→

```
3. import java.sql.*;
4.
5. public class jdbc {
6.     public static void main(String[] args) {
7.         try {
8.             // Establish a connection to the database
9.             Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost:5434/postgres",
"postgres", "saqlain");
10.
11.             // Prepare a statement to insert data into the Teacher table
12.             PreparedStatement insertStmt = conn.prepareStatement("INSERT
INTO Teacher (Tno, Tname, Subject) VALUES (?, ?, ?)");
13.
14.             // Insert at least 5 records into the Teacher table
15.             insertStmt.setInt(1, 1);
16.             insertStmt.setString(2, "John Doe");
17.             insertStmt.setString(3, "JAVA");
18.             insertStmt.executeUpdate();
19.
20.             insertStmt.setInt(1, 2);
21.             insertStmt.setString(2, "Jane Smith");
22.             insertStmt.setString(3, "Python");
23.             insertStmt.executeUpdate();
24.
25.             insertStmt.setInt(1, 3);
26.             insertStmt.setString(2, "Mike Johnson");
27.             insertStmt.setString(3, "C++");
28.             insertStmt.executeUpdate();
29.
30.             insertStmt.setInt(1, 4);
31.             insertStmt.setString(2, "Sarah Lee");
32.             insertStmt.setString(3, "JAVA");
33.             insertStmt.executeUpdate();
34.
35.             insertStmt.setInt(1, 5);
36.             insertStmt.setString(2, "David Chen");
37.             insertStmt.setString(3, "PHP");
38.             insertStmt.executeUpdate();
39.
40.             // Prepare a statement to retrieve data from the Teacher table
```

```

41.         PreparedStatement selectStmt = conn.prepareStatement("SELECT *
FROM Teacher WHERE Subject = ?");
42.
43.         // Retrieve the details of the teacher who is teaching "JAVA"
Subject
44.         selectStmt.setString(1, "JAVA");
45.         ResultSet rs = selectStmt.executeQuery();
46.         while (rs.next()) {
47.             int tno = rs.getInt("Tno");
48.             String tname = rs.getString("Tname");
49.             String subject = rs.getString("Subject");
50.             System.out.println("Teacher number: " + tno + ", Teacher
name: " + tname + ", Subject: " + subject);
51.         }
52.
53.         // Close the resources
54.         rs.close();
55.         selectStmt.close();
56.         insertStmt.close();
57.         conn.close();
58.     } catch (SQLException e) {
59.         e.printStackTrace();
60.     }
61. }
62. }

```

Slip no 17:

1:o/p→

```
import java.util.*;
```

```

public class SortedIntegers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of integers: ");
        int n = scanner.nextInt();
        Set<Integer> set = new TreeSet<>();
        for (int i = 1; i <= n; i++) {
            System.out.print("Enter integer #" + i + ": ");
            int num = scanner.nextInt();
            set.add(num);
        }
    }
}

```

```

        System.out.println("The integers in sorted order are: ");
        for (int num : set) {
            System.out.print(num + " ");
        }
    }
}

```

2.o/p→

```

import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class NumberDisplay implements Runnable {
    private JTextField textField;

    public NumberDisplay(JTextField textField) {
        this.textField = textField;
    }

    @Override
    public void run() {
        for (int i = 1; i <= 100; i++) {

```

```

        textField.setText(Integer.toString(i));
    try {
        Thread.sleep(100);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}

```

```

public static void main(String[] args) {
    JFrame frame = new JFrame("Number Display");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JTextField textField = new JTextField(10);
    JButton button = new JButton("Start");
    JPanel panel = new JPanel();
    panel.add(textField);
    panel.add(button);
    frame.add(panel, BorderLayout.CENTER);
    frame.pack();
    frame.setVisible(true);
    button.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            NumberDisplay numberDisplay = new NumberDisplay(textField);
            Thread thread = new Thread(numberDisplay);
            thread.start();
        }
    });
}

```

```
    }  
    });  
}  
}
```

Slip no 18:

1.o/p→

```
public class ThreadInfo {  
    public static void main(String[] args) {  
        Thread thread = Thread.currentThread();  
        System.out.println("Thread Name: " + thread.getName());  
        System.out.println("Thread Priority: " + thread.getPriority());  
    }  
}
```

2.o/p→

```
import java.io.IOException;  
import java.io.PrintWriter;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
public class StudentDetails extends HttpServlet {  
    private static final long serialVersionUID = 1L;
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
    throws ServletException, IOException {  
    response.setContentType("text/html");  
    PrintWriter out = response.getWriter();
```

```
  
    String seatNo = request.getParameter("seatno");  
    String studName = request.getParameter("studname");  
    String studentClass = request.getParameter("class");  
    int totalMarks = Integer.parseInt(request.getParameter("totalmarks"));  
    double percentage = (double) totalMarks / 5;
```

```
  
    String grade;  
    if (percentage >= 80) {  
        grade = "A+";  
    } else if (percentage >= 70) {  
        grade = "A";  
    } else if (percentage >= 60) {  
        grade = "B";  
    } else if (percentage >= 50) {  
        grade = "C";  
    } else {  
        grade = "F";  
    }  
}
```

```
  
    out.println("<html>");  
    out.println("<head>");
```

```

        out.println("<title>Student Details</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Student Details</h1>");
        out.println("<table>");
        out.println("<tr><td>Seat No:</td><td>" + seatNo + "</td></tr>");
        out.println("<tr><td>Student Name:</td><td>" + studName +
"</td></tr>");
        out.println("<tr><td>Class:</td><td>" + studentClass + "</td></tr>");
        out.println("<tr><td>Total Marks:</td><td>" + totalMarks + "</td></tr>");
        out.println("<tr><td>Percentage:</td><td>" + percentage + "%</td></tr>");
        out.println("<tr><td>Grade:</td><td>" + grade + "</td></tr>");
        out.println("</table>");
        out.println("</body>");
        out.println("</html>");
    }
}

```

Slip no 19:

1.o/p→

```
import java.util.LinkedList;
```

```
import java.util.Scanner;
```

```

public class NegativeIntegersInLinkedList {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        LinkedList<Integer> list = new LinkedList<Integer>();
    }
}

```

```
System.out.print("Enter the number of integers: ");  
int n = sc.nextInt();
```

```
System.out.println("Enter " + n + " integers:");  
for (int i = 0; i < n; i++) {  
    int num = sc.nextInt();  
    list.add(num);  
}
```

```
System.out.println("Negative Integers:");  
for (int i : list) {  
    if (i < 0) {  
        System.out.println(i);  
    }  
}  
}  
}
```

2.o/p→

```
import java.io.IOException;  
import java.io.PrintWriter;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;
```



```
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String username = request.getParameter("username");
        String password = request.getParameter("password");

        try {
            // Load the JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Set up the connection
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase",
"root", "password");
```

```

        // Prepare the statement
        PreparedStatement stmt = conn.prepareStatement("SELECT * FROM
users WHERE username=? AND password=?");

        stmt.setString(1, username);
        stmt.setString(2, password);


        // Execute the query
        ResultSet rs = stmt.executeQuery();


        // Check if the user exists
        if (rs.next()) {
            out.println("<h1>Login successful!</h1>");
        } else {
            out.println("<h1>Error: Invalid username or password.</h1>");
        }


        // Close the resources
        rs.close();
        stmt.close();
        conn.close();
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
    }
}
}

```

Slip no 20:

1.o/p→

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Number to Words Converter</title>

<style>

    .red {
        color: red;
    }

</style>

</head>

<body>

    <h1>Number to Words Converter</h1>

    <form method="post">

        Enter a number: <input type="text" name="number" /><br />

        <input type="submit" value="Convert" />

    </form>

    <br />

    <%-- Get the number from the request parameter --%>

    <% String numberStr = request.getParameter("number"); %>

    <%-- Check if the number is not null and not empty --%>

    <% if (numberStr != null && !numberStr.trim().isEmpty()) { %>

        <%-- Convert the number to words --%>
```

```

    <% String[] words = { "Zero", "One", "Two", "Three", "Four", "Five", "Six",
"Seven", "Eight", "Nine" }; %>

    <% String number = numberStr.trim(); %>

    <span class="red">

        <% for (int i = 0; i < number.length(); i++) { %>

            <%= words[Character.getNumericValue(number.charAt(i))] + " " %>

            <% } %>

        </span>

    <% } %>
</body>
</html>

```

2.o/p→

```

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.Timer;

```

```

public class BlinkingImage extends JPanel implements ActionListener {

```

```
private static final long serialVersionUID = 1L;
```

```
private Image image;
```

```
private boolean blinkOn = true;
```

```
public BlinkingImage(Image image) {
```

```
    this.image = image;
```

```
    Timer timer = new Timer(500, this); // timer fires every 500ms
```

```
    timer.start();
```

```
}
```

```
@Override
```

```
protected void paintComponent(Graphics g) {
```

```
    super.paintComponent(g);
```

```
    if (blinkOn) {
```

```
        g.drawImage(image, 0, 0, this);
```

```
    }
```

```
}
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    blinkOn = !blinkOn;
```

```
    repaint();
```

```
}
```

```
public static void main(String[] args) {
```

```
    try {
```

```

        Image image = ImageIO.read(new File("path/to/image.png"));
        JPanel contentPane = new BlinkingImage(image);
        contentPane.setPreferredSize(new Dimension(image.getWidth(null),
image.getHeight(null)));
        JFrame frame = new JFrame("Blinking Image");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().add(contentPane, BorderLayout.CENTER);
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Slip no 21:

1.o/p→

```

import java.util.Iterator;
import java.util.LinkedList;
import java.util.Scanner;

```

```

public class SubjectNames {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of subjects: ");
        int n = sc.nextInt();
    }
}

```

```

LinkedList<String> subjects = new LinkedList<String>();
System.out.println("Enter the subject names:");
for (int i = 0; i < n; i++) {
    String subject = sc.next();
    subjects.add(subject);
}

```

```

System.out.println("Subject names:");
Iterator<String> iterator = subjects.iterator();
while (iterator.hasNext()) {
    String subject = iterator.next();
    System.out.println(subject);
}
}
}

```

2.o/p→

```
import java.util.LinkedList;
```

```

public class ProducerConsumer {
    public static void main(String[] args) {
        LinkedList<Integer> buffer = new LinkedList<>();
        int bufferSize = 5;
        Thread producerThread = new Thread(new Producer(buffer, bufferSize),
"Producer");
        Thread consumerThread = new Thread(new Consumer(buffer),
"Consumer");
    }
}

```

```
    producerThread.start();
    consumerThread.start();
}
}
```

```
class Producer implements Runnable {
```

```
    private final LinkedList<Integer> buffer;
```

```
    private final int bufferSize;
```

```
    private int value = 0;
```

```
    public Producer(LinkedList<Integer> buffer, int bufferSize) {
```

```
        this.buffer = buffer;
```

```
        this.bufferSize = bufferSize;
```

```
    }
```

```
    @Override
```

```
    public void run() {
```

```
        while (true) {
```

```
            synchronized (buffer) {
```

```
                if (buffer.size() < bufferSize) {
```

```
                    buffer.add(value);
```

```
                    System.out.println(Thread.currentThread().getName() + " produced "
+ value);
```

```
                    value++;
```

```
                    buffer.notifyAll();
```

```
                } else {
```

```
                    try {
```



```

        buffer.wait();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
}
}
}
}
}
}
}
}

```

```

class Consumer implements Runnable {
    private final LinkedList<Integer> buffer;

    public Consumer(LinkedList<Integer> buffer) {
        this.buffer = buffer;
    }
}

```

```

@Override
public void run() {
    while (true) {
        synchronized (buffer) {
            if (buffer.isEmpty()) {
                try {
                    buffer.wait();
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

```

        }
    } else {
        int value = buffer.removeFirst();
        System.out.println(Thread.currentThread().getName() + " consumed
" + value);
        buffer.notifyAll();
    }
}
}
}
}
}
}
}

```

Slip no 22:

1.o/p→

```
import java.sql.*;
```

```

public class EmployeeManagement {
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/mydatabase";
    static final String USER = "username";
    static final String PASSWORD = "password";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try {
            Class.forName(JDBC_DRIVER);

```

```
conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);
stmt = conn.createStatement();
int choice;
do {
    System.out.println("Menu:");
    System.out.println("1. Insert");
    System.out.println("2. Update");
    System.out.println("3. Display");
    System.out.println("4. Exit");
    System.out.print("Enter your choice: ");
    choice = Integer.parseInt(System.console().readLine());
    switch (choice) {
        case 1:
            System.out.print("Enter employee number: ");
            int eno = Integer.parseInt(System.console().readLine());
            System.out.print("Enter employee name: ");
            String ename = System.console().readLine();
            System.out.print("Enter employee salary: ");
            double salary = Double.parseDouble(System.console().readLine());
            String sql = "INSERT INTO Employee (ENo, EName, Salary) VALUES
(" + eno + ", '" + ename + "', " + salary + ")";
            stmt.executeUpdate(sql);
            System.out.println("Record inserted successfully.");
            break;
        case 2:
            System.out.print("Enter employee number: ");
            eno = Integer.parseInt(System.console().readLine());
```

```

        System.out.print("Enter new employee name: ");
        ename = System.console().readLine();
        System.out.print("Enter new employee salary: ");
        salary = Double.parseDouble(System.console().readLine());
        sql = "UPDATE Employee SET EName='" + ename + "', Salary=" +
salary + " WHERE ENo=" + eno;

        int rowsAffected = stmt.executeUpdate(sql);
        if (rowsAffected == 0) {
            System.out.println("No records found with employee number "
+ eno);
        } else {
            System.out.println(rowsAffected + " record(s) updated
successfully.");
        }
        break;
    case 3:
        sql = "SELECT * FROM Employee";
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            eno = rs.getInt("ENo");
            ename = rs.getString("EName");
            salary = rs.getDouble("Salary");

            System.out.println("Employee number: " + eno + ", Employee
name: " + ename + ", Employee salary: " + salary);
        }
        rs.close();
        break;

```

```

        case 4:
            System.out.println("Exiting program.");
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
            break;
    }
    } while (choice != 4);
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        if (stmt != null) {
            stmt.close();
        }
        if (conn != null) {
            conn.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

2.o/p→

<%@ page language="java" %>

```
<html>
```

```
<head>
```

```
<title>Greeting Page</title>
```

```
</head>
```

```
<body>
```

```
<h1>Welcome to our website</h1>
```

```
<%
```

```
String userName = request.getParameter("userName"); // retrieve user input  
from the form
```

```
String greetingMessage = "Hello, " + userName;
```

```
java.util.Date date = new java.util.Date(); // get the current date and time
```

```
java.text.SimpleDateFormat sdf = new java.text.SimpleDateFormat("hh:mm:ss  
a"); // format the date and time
```

```
int hour = Integer.parseInt(sdf.format(date).substring(0, 2)); // extract the hour  
from the formatted date
```

```
if (hour >= 0 && hour < 12) {
```

```
    greetingMessage += ". Good morning!";
```

```
} else if (hour >= 12 && hour < 18) {
```

```
    greetingMessage += ". Good afternoon!";
```

```
} else {
```

```
    greetingMessage += ". Good evening!";
```

```
}
```

```
%>
```

```
<p><%= greetingMessage %></p>
```

```
<form action="greeting.jsp" method="get">
```

```
Enter your name: <input type="text" name="userName">
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

Slip no 23:

1.o/p→

```
import java.util.Scanner;
```

```
public class VowelDisplay {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter a string: ");
```

```
        String inputString = scanner.nextLine().toLowerCase(); // convert input to lowercase
```

```
        for (int i = 0; i < inputString.length(); i++) {
```

```
            char c = inputString.charAt(i);
```

```
            if (isVowel(c)) {
```

```
                System.out.print(c + " ");
```

```
                try {
```

```
                    Thread.sleep(3000); // pause for 3 seconds
```

```

        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
}

```

```

public static boolean isVowel(char c) {
    return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
}
}

```

2.o/p→

```

import java.util.ArrayList;
import java.util.Iterator;
import java.util.ListIterator;

```

```

public class StudentNames {
    public static void main(String[] args) {
        ArrayList<String> studentNames = new ArrayList<>();

        // Accept N student names through command line and add them to the
        ArrayList
        for (int i = 0; i < args.length; i++) {
            studentNames.add(args[i]);
        }

        // Display the student names using Iterator interface
    }
}

```



```

System.out.println("Student names using Iterator:");
Iterator<String> iterator = studentNames.iterator();
while (iterator.hasNext()) {
    System.out.println(iterator.next());
}

// Display the student names using ListIterator interface
System.out.println("\nStudent names using ListIterator:");
ListIterator<String> listIterator = studentNames.listIterator();
while (listIterator.hasNext()) {
    System.out.println(listIterator.next());
}
}
}

```

Slip no 24:

1.o/p→

```

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class ScrollingText extends JPanel implements Runnable {
    private int x, y;
    private String message;
    private Thread thread;
}

```

```
private boolean running;
```

```
public ScrollingText(String message) {  
    this.message = message;  
    x = getWidth();  
    y = getHeight() / 2;  
    setFont(new Font("Arial", Font.BOLD, 20));  
    setForeground(Color.RED);  
    setBackground(Color.WHITE);  
    thread = new Thread(this);  
    running = true;  
    thread.start();  
}
```

```
public void run() {  
    while (running) {  
        try {  
            Thread.sleep(20); // Adjust the speed of scrolling here  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
        x--;  
        if (x < -getFontMetrics(getFont()).stringWidth(message)) {  
            x = getWidth();  
        }  
        repaint();  
    }  
}
```

```
}  
}
```

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    g.drawString(message, x, y);  
}
```

```
public static void main(String[] args) {  
    JFrame frame = new JFrame("Scrolling Text");  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.setSize(400, 100);  
    ScrollingText scrollingText = new ScrollingText("This is scrolling text");  
    frame.add(scrollingText);  
    frame.setVisible(true);  
}  
}
```

2.o/p→

```
<%@ page language="java" %>
```

```
<%@ page import="java.io.*, java.util.*" %>
```

```
<%
```

```
String username = request.getParameter("username");
```

```
String password = request.getParameter("password");
```

```
if (username.equals(password)) {
```

```

        response.sendRedirect("Login.html?message=Login Successfully");
    } else {
        response.sendRedirect("Error.html?message=Login Failed");
    }
%>

```

Slip no 25:

1.o/p→

```
<%@ page language="java" %>
```

```
<%@ page import="java.io.*, java.util.*" %>
```

```
<%
```

```
    String name = request.getParameter("name");
```

```
    int age = Integer.parseInt(request.getParameter("age"));
```

```
    if (age >= 18) {
```

```
        out.println("<h3>" + name + ", you are eligible to vote.</h3>");
```

```
    } else {
```

```
        out.println("<h3>" + name + ", you are not eligible to vote.</h3>");
```

```
    }
```

```
%>
```

```
<form method="post" action="">
```

```
    <label for="name">Name:</label>
```

```
    <input type="text" name="name" id="name" required><br><br>
```

```
    <label for="age">Age:</label>
```

```
<input type="number" name="age" id="age" required><br><br>
```

```
<input type="submit" value="Check Eligibility">
```

```
</form>
```

```
2.o/p→
```

Leave

Slip no 26:

```
1.o/p→
```

```
import java.sql.*;
```

```
public class DeleteEmployeeDetails {
```

```
    public static void main(String[] args) {
```

```
        try {
```

```
            // Step 1: Load the JDBC driver
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            // Step 2: Establish the connection to the database
```

```
            String url = "jdbc:mysql://localhost:3306/yourdatabase";
```

```
            String username = "yourusername";
```

```
            String password = "yourpassword";
```

```
            Connection con = DriverManager.getConnection(url, username,  
password);
```

```
            // Step 3: Create a prepared statement to delete the employee details
```

```
            PreparedStatement pstmt = con.prepareStatement("DELETE FROM  
Employee WHERE ENo=?");
```

```
// Step 4: Set the employee ID parameter for the prepared statement
```

```
int employeeID = Integer.parseInt(args[0]);
```

```
pstmt.setInt(1, employeeID);
```

```
// Step 5: Execute the prepared statement
```

```
int rowsAffected = pstmt.executeUpdate();
```

```
System.out.println(rowsAffected + " row(s) deleted.");
```

```
// Step 6: Close the resources
```

```
pstmt.close();
```

```
con.close();
```

```
} catch (Exception e) {
```

```
    e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

2.o/p→

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Sum of First and Last Digit</title>
```

```
</head>
```

```
<body>
```

```
    <h1>Calculate Sum of First and Last Digit</h1>
```

```
    <%
```

```
        // Retrieve the number from the request parameter
```

```

int num = Integer.parseInt(request.getParameter("number"));

// Extract the first and last digit of the number
int firstDigit = num;
while (firstDigit >= 10) {
    firstDigit /= 10;
}
int lastDigit = num % 10;

// Calculate the sum of first and last digit
int sum = firstDigit + lastDigit;

// Display the sum in red color with font size 18
out.println("<p style=\"color:red;font-size:18px;\">Sum of first and
last digit of " + num + " is " + sum + "</p>");
%>
<form action="" method="get">
    <label for="number">Enter a number:</label>
    <input type="number" id="number" name="number" required>
    <button type="submit">Calculate</button>
</form>
</body>
</html>

```

Slip no 27:

1.o/p→

```
import java.sql.*;
```

```
import javax.swing.*;

import javax.swing.table.DefaultTableModel;

public class CollegeDetailsJTable extends JFrame {

    // database connection details
    static final String DB_URL = "jdbc:mysql://localhost:3306/college";
    static final String DB_USER = "root";
    static final String DB_PASSWORD = "password";

    public CollegeDetailsJTable() {
        super("College Details");

        // create a JTable
        JTable table = new JTable();

        // set column names for the table
        String[] columnNames = {"CID", "CName", "Address", "Year"};

        // create a default table model with the column names
        DefaultTableModel model = new DefaultTableModel(columnNames, 0);

        // add data to the table model
        try {
            // create a database connection
            Connection conn = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD);
```



```
// create a statement
Statement stmt = conn.createStatement();

// execute a query to get college details
ResultSet rs = stmt.executeQuery("SELECT * FROM college");

// add each row to the table model
while (rs.next()) {
    String cid = rs.getString("CID");
    String cname = rs.getString("CName");
    String address = rs.getString("Address");
    int year = rs.getInt("Year");
    model.addRow(new Object[] { cid, cname, address, year });
}

// close the result set, statement and connection
rs.close();
stmt.close();
conn.close();
} catch (SQLException e) {
    e.printStackTrace();
}

// set the table model to the JTable
table.setModel(model);
```

```

// create a scroll pane for the table
JScrollPane scrollPane = new JScrollPane(table);

// add the scroll pane to the frame
getContentPane().add(scrollPane);

// set frame properties
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);
pack();
setVisible(true);
}

```

```

public static void main(String[] args) {
    new CollegeDetailsJTable();
}
}

```

2.o/p→

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

```

```

public class ChangeSessionTimeout extends HttpServlet {

```

```

    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

```

```
// Get the current session
HttpSession session = request.getSession();

// Set the inactive time interval to 10 minutes (in seconds)
session.setMaxInactiveInterval(600);

// Display a message
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html><body>");
out.println("<h3>Session Timeout Changed to 10 Minutes</h3>");
out.println("</body></html>");
}

}
```

Slip no 28:

1.o/p→

```
<%@ page language="java" %>
<html>
<head>
    <title>Reverse String</title>
</head>
<body>
    <h2>Enter a string to reverse:</h2>
    <form method="post" action="reverse.jsp">
```

```

        <input type="text" name="stringToReverse">
        <input type="submit" value="Reverse">
    </form>
    <%
        String inputString = request.getParameter("stringToReverse");
        if (inputString != null) {
            String reversedString = "";
            for (int i = inputString.length() - 1; i >= 0; i--) {
                reversedString += inputString.charAt(i);
            }
            out.print("<h2>Reversed string: " + reversedString + "</h2>");
        }
    %>
</body>
</html>
2.o/p→
public class CurrentThreadName {
    public static void main(String[] args) {
        Thread t = Thread.currentThread();
        System.out.println("Currently executing thread: " + t.getName());
    }
}

```

Slip no 29:

```

1.o/p→
import java.sql.*;

```

```

public class DonarTableColumnInfo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/mydb"; // replace mydb with your
        database name

        String username = "root"; // replace root with your MySQL username

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

        try (Connection con = DriverManager.getConnection(url, username,
        password)) {
            String query = "SELECT * FROM DONAR";
            PreparedStatement ps = con.prepareStatement(query);
            ResultSet rs = ps.executeQuery();

            ResultSetMetaData rsmd = rs.getMetaData();
            int columnCount = rsmd.getColumnCount();

            System.out.println("Column Name\t\tData Type");
            System.out.println("-----");

            for (int i = 1; i <= columnCount; i++) {
                String columnName = rsmd.getColumnName(i);
                String dataType = rsmd.getColumnTypeName(i);
                System.out.println(columnName + "\t\t" + dataType);
            }
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

```
    }  
    }  
}
```

2.o/p→

```
import java.util.LinkedList;
```

```
public class LinkedListDemo {  
    public static void main(String[] args) {  
        // Create a LinkedList of integers  
        LinkedList<Integer> list = new LinkedList<>();  
  
        // Add element at the first position  
        list.addFirst(10);  
        list.addFirst(20);  
        list.addFirst(30);  
  
        System.out.println("LinkedList after adding elements at first position: " +  
list);  
  
        // Delete the last element  
        list.removeLast();  
  
        System.out.println("LinkedList after deleting last element: " + list);  
  
        // Display the size of LinkedList  
        System.out.println("Size of LinkedList: " + list.size());  
    }  
}
```

Slip no 30:

1.o/p→

```
class Counter {  
    private int count = 0;  
  
    // synchronized method  
    public synchronized void increment() {  
        count++;  
    }  
  
    public int getCount() {  
        return count;  
    }  
}
```

```
class MyThread extends Thread {  
    private Counter counter;  
  
    public MyThread(Counter counter) {  
        this.counter = counter;  
    }  
  
    public void run() {  
        for (int i = 0; i < 1000; i++) {  
            counter.increment();  
        }  
    }  
}
```

```
}  
}
```

```
public class SyncDemo {  
    public static void main(String[] args) throws InterruptedException {  
        Counter counter = new Counter();  
  
        MyThread thread1 = new MyThread(counter);  
        MyThread thread2 = new MyThread(counter);  
  
        thread1.start();  
        thread2.start();  
  
        thread1.join();  
        thread2.join();  
  
        System.out.println("Count: " + counter.getCount());  
    }  
}
```

2.o/p→

```
import java.sql.*;  
  
public class ScrollableResultSetExample {  
  
    public static void main(String[] args) {
```



```
try {  
    // Register JDBC driver  
    Class.forName("com.mysql.jdbc.Driver");  
  
    // Open a connection  
    Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb",  
"username", "password");  
  
    // Create a Statement object with scrollable ResultSet  
    Statement stmt =  
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,  
ResultSet.CONCUR_READ_ONLY);  
  
    // Execute a query and get the ResultSet object  
    ResultSet rs = stmt.executeQuery("SELECT * FROM Teacher");  
  
    // Move the cursor to the last row  
    rs.last();  
  
    // Get the total number of rows in the ResultSet  
    int rowCount = rs.getRow();  
  
    // Move the cursor back to the first row  
    rs.beforeFirst();  
  
    // Display the column headers
```

```
ResultSetMetaData rsmd = rs.getMetaData();
for (int i = 1; i <= rsmd.getColumnCount(); i++) {
    System.out.print(rsmd.getColumnName(i) + "\t");
}
System.out.println();

// Iterate through the ResultSet and display the data
while (rs.next()) {
    System.out.print(rs.getInt("TID") + "\t");
    System.out.print(rs.getString("TName") + "\t");
    System.out.print(rs.getInt("Salary") + "\t");
    System.out.println();
}

// Close the ResultSet, Statement, and Connection
rs.close();
stmt.close();
conn.close();

} catch (Exception ex) {
    ex.printStackTrace();
}
}
```