

Slip1

1. Write a Java program to display all the alphabets between 'A' to 'Z' after every 2 seconds. [15 M]

```
public class Slip1_1 extends Thread
{
    char ch;

    public void run()
    {
        for(ch = 'A'; ch<='Z';ch++)
        {
            System.out.println(ch+" ");
            try
            {
                Thread.sleep(2000);
            } catch(Exception e){}
        }
    }

    public static void main(String args[])
    {
        Slip1_1 t = new Slip1_1();
        t.start();
    }
}
```

2. Write a Java program to accept the details of Employee (Eno, EName, Designation, Salary) from a user and store it into the database. (Use Swing) [15 M]

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

class Slip1_2_JDBC extends JFrame implements ActionListener
{

```

```

JLabel l1,l2,l3,l4;

JTextField t1,t2,t3,t4;

JButton btnAdd;

Connection con=null;

ResultSet rs=null;

Slip1_2_JDBC()
{
    l1= new JLabel("Employee no:");
    l2= new JLabel("Employee Name:");
    l3= new JLabel("Employee Designation:");
    l4= new JLabel("Employee Salary:");
    t1= new JTextField();
    t2= new JTextField();
    t3= new JTextField();
    t4= new JTextField();
    btnAdd = new JButton("Add");

    setLayout(new FlowLayout());
    setTitle("Employee Registration");
    setSize(350,250);

    Panel p1=new Panel();
    p1.setLayout(new GridLayout(4,2,20,20));
    p1.add(l1);p1.add(t1);
    p1.add(l2);p1.add(t2);
    p1.add(l3);p1.add(t3);
    p1.add(l4);p1.add(t4);

    add(p1);
    add(btnAdd);

    try{

        Class.forName("org.postgresql.Driver");

        con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

```

```

    }

    catch(Exception e)

    {
System.out.println(e);
    }

    btnAdd.addActionListener(this);

    setVisible(true);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    public void actionPerformed(ActionEvent ae)
    {
        int no=Integer.parseInt(t1.getText());
        String name=t2.getText();
        String desg=t3.getText();
        int sal=Integer.parseInt(t4.getText());
        try{
            PreparedStatement ps=con.prepareStatement("insert into employee
values(?,?,?,?)");

            ps.setInt(1,no);

            ps.setString(2,name);

            ps.setString(3,desg);

            ps.setInt(4,sal);

            int i=ps.executeUpdate();

            JOptionPane.showMessageDialog(null,"Record Inserted Succfully");
        }catch(Exception e){}

    }

    public static void main(String arg[])
    {
        new Slip1_2_JDBC();
    }

```

```
}
```

Slip2

1. Write a java program to read 'N' names of your friends, store it into HashSet and display them in ascending order. [15 M]

```
import java.util.*;

public class Slip2_1
{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        HashSet<String> names = new HashSet<>();
        System.out.print("Enter the number of friends: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name: ");
            names.add(scanner.nextLine());
        }

        TreeSet<String> sortedNames = new TreeSet<>(names);
        for (String name : sortedNames)
            System.out.println(name);
        scanner.close();
    }
}
```

2. Design a servlet that provides information about a HTTP request from a client, such as IP-Address and browser type. The servlet also provides information about the server on which the servlet is running, such as the operating system type, and the names of currently loaded servlets. [15 M]

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.util.*;

public class Slip2_2 extends HttpServlet

{

    public void doGet(HttpServletRequest request, HttpServletResponse response)

    {

        try

        {

            response.setContentType("text/html");

            PrintWriter out = response.getWriter();

            out.println("<html>");

            out.println("<body>");

            out.println("<head>");

            out.println("<title>Request Information Example</title>");

            out.println("</head>");

            out.println("<body>");

            out.println("<h3>Request Information Example</h3>");

            java.util.Properties p=System.getProperties();

                String serverOs = System.getProperty("os.name");

            out.println("Server Name :"+request.getServerName()+"<br>");

            out.println("Operating System Name :"+serverOs+"<br>");

            out.println("Method: " + request.getMethod()+"<Br>");

            out.println("Request URI: " + request.getRequestURI()+"<Br>");

            out.println("Protocol: " + request.getProtocol()+"<Br>");

            out.println("PathInfo: " +request.getServletPath()+"<Br>");

            out.println("Remote Address: " +request.getRemoteAddr()+"<Br>");
```

```

        out.println("</body>");
        out.println("</html>");
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}

```

Slip3

1. Write a JSP program to display the details of Patient (PNo, PName, Address, age, disease) in tabular form on browser. [15 M]

```

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

<h2>Patient Details</h2>

<table border="1">

    <tr>

        <th>Patient Number</th>

        <th>Name</th>

        <th>Address</th>

        <th>Age</th>

        <th>Disease</th>

    </tr>

    <%

    try {

        Class.forName("org.postgresql.Driver");

```

```

Connection c =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

        Statement st=c.createStatement();

        ResultSet rs = st.executeQuery("SELECT * FROM Patient");

        while(rs.next()) {
%>

<tr>

        <td><%= rs.getInt(1) %></td>

        <td><%= rs.getString(2) %></td>

        <td><%= rs.getString(3) %></td>

        <td><%= rs.getInt(4) %></td>

        <td><%= rs.getString(5) %></td>

</tr>

<%

        }

        rs.close();

        st.close();

        c.close();

    } catch (Exception e) {

        e.printStackTrace();

    }

%>

</table>

```

2. Write a Java program to create LinkedList of String objects and perform the following: i. Add element at the end of the list ii. Delete first element of the list iii. Display the contents of list in reverse order [15 M]

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

```
import java.util.ListIterator;
```

```
public class Slip3_2 {
```

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

```
        LinkedList<String> list = new LinkedList<>();
```

```

        list.add("Apple");

        list.add("Banana");

        list.add("Cherry");

System.out.println("Original List : "+list);

        list.addLast("Date");

System.out.println("After Adding element at last : "+list);

        list.removeFirst();

System.out.println("After removing first element : "+list);

System.out.println("List elements in reverse order ");

        ListIterator<String> iterator = list.listIterator(list.size());

        while (iterator.hasPrevious()) {

            System.out.println(iterator.previous());

        }

    }

}

```

Slip4

1. Write a Java program using Runnable interface to blink Text on the frame. [15 M]

```

import java.awt.*;

import javax.swing.*;

import java.awt.event.*;

class Slip4_1 extends JFrame implements Runnable

{

    Thread t;

    JLabel l1;

    int flag;

    Slip4_1()

```



```
{
t=new Thread(this);
t.start();
setLayout(null);
l1=new JLabel("This is Multithreading");
l1.setBounds(100,100,150,40);
add(l1);
setSize(300,300);
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
flag=0;
}
public void run()
{
try
{
if(flag==0)
{
t.sleep(20);
l1.setText("");
flag=1;
}
if(flag==1)
{
t.sleep(20);
l1.setText("This is Multithreading");
flag=0;
}
}
catch(Exception e)
{

```

```

System.out.println(e);
} run();
}
public static void main(String a[])
{
new Slip4_1();
}}

```

2. Write a Java program to store city names and their STD codes using an appropriate collection and perform following operations: i. Add a new city and its code (No duplicates) ii. Remove a city from the collection iii. Search for a city name and display the code [15 M]

```

import java.util.*;
import java.io.*;
class Slip4_2
{
public static void main(String args[]) throws IOException
{
    Hashtable ht=new Hashtable();
    int std_code;
    String city_name=null;
Scanner sc=new Scanner(System.in);
    System.out.println("\n Enter no of Records : ");
    int n=sc.nextInt();
    for(int i=0;i<n;i++)
    {
        System.out.print("Enter City name :");
        city_name=sc.next();
        System.out.print("Enter STD code :");
        std_code = sc.nextInt();
        ht.put(city_name,std_code);
    }
    System.out.println("Hash table = "+ht);
}
}

```

```

System.out.println("Enter city name to be search : ");
String search_nm=sc.next();
if(ht.containsKey(search_nm))
    System.out.println("STD Code of City "+search_nm+" is"+ht.get(search_nm));
else
    System.out.println("City not found");
System.out.println("Enter city name to be remove : ");
String removeCity=sc.next();
ht.remove(removeCity);
System.out.println("After removing city "+ removeCity + " data is "+ht);
}}

```

Slip5

1. Write a Java Program to create the hash table that will maintain the mobile number and student name. Display the details of student using Enumeration interface. [15 M]

```

import java.util.Map.Entry;
import java.util.*;
public class HashtableDemoSetA4
{
    public static void main(String args[])
    {
        Scanner s =new Scanner(System.in);
        System.out.println("Enter no. of students you want");
        int n=s.nextInt();
        Hashtable<String, Long> contact = new Hashtable<String, Long>();
        for(int i=0;i<n;i++)
        {

```

```

System.out.println("Enter student's Name");

String name=s.next();

System.out.println("Enter student's mobile number");

long no=s.nextLong();

contact.put(name, no);

}

System.out.println("Map = "+contact);

Set s1=contact.entrySet();

Iterator i=s1.iterator();

System.out.println("Name    Contact Number");

while(i.hasNext())

{

    Map.Entry me=(Entry) i.next();

    System.out.println(""+me.getKey()+"    "+me.getValue());

}}

}

```

2. Create a JSP page for an online multiple choice test. The questions are randomly selected from a database and displayed on the screen. The choices are displayed using radio buttons. When the user clicks on next, the next question is displayed. When the user clicks on submit, display the total score on the screen. [15 M]

//OnlineTest.jsp

```

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

<html>

<head>

    <title>Online Quiz</title>

</head>

<body>

    <h2>Online Multiple Choice Test</h2>

    <form action="result.jsp" method="post">

        <%

```

```

Class.forName("org.postgresql.Driver");

Connection conn =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb", "postgres", "123");

Statement stmt = conn.createStatement();

// Select 5 random questions
String sql = "SELECT * FROM questions ORDER BY RANDOM() LIMIT 5";
ResultSet rs = stmt.executeQuery(sql);
int i = 1;

while (rs.next()) {
    int id = rs.getInt("id");

    String question = rs.getString("question");

    String option1 = rs.getString("option1");
    String option2 = rs.getString("option2");
    String option3 = rs.getString("option3");
    String option4 = rs.getString("option4");

    out.println("<p>" + i + ". " + question + "</p>");
    out.println("<input type='hidden' name='q' + i + \"' value='\" + id + \"'>");
    out.println("<input type='radio' name='ans\" + i + \"' value='1'>" + option1 + "<br>");
    out.println("<input type='radio' name='ans\" + i + \"' value='2'>" + option2 + "<br>");
    out.println("<input type='radio' name='ans\" + i + \"' value='3'>" + option3 + "<br>");
    out.println("<input type='radio' name='ans\" + i + \"' value='4'>" + option4 + "<br><br>");
    i++;
}

rs.close();
stmt.close();
conn.close();

%>

```

```

        <input type="submit" value="Submit">
    </form>
</body>
</html>
//result.jsp
<%@ page import="java.sql.*, java.util.*" %>
<html>
<head>
    <title>Quiz Result</title>
</head>
<body>
    <h2>Quiz Result</h2>
    <%
        int score = 0;

        Class.forName("org.postgresql.Driver");

        Connection conn =
        DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb", "postgres", "123");

        PreparedStatement pstmt = conn.prepareStatement("SELECT correct_option FROM
        questions WHERE id=?");

        for (int i = 1; i <= 5; i++) {

            String questionId = request.getParameter("q" + i);

            String answer = request.getParameter("ans" + i);

            if (questionId != null && answer != null) {

                pstmt.setInt(1, Integer.parseInt(questionId));

                ResultSet rs = pstmt.executeQuery();

                if (rs.next()) {

                    int correctOption = rs.getInt("correct_option");

                    if (correctOption == Integer.parseInt(answer)) {

                        score++;

```

```

        }
    }
    rs.close();
}
}

pstmt.close();
conn.close();

out.println("<h3>Your Score: " + score + " / 5</h3>");
%>
</body>
</html>

```

Slip6

1. Write a Java program to accept 'n' integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection). Search for a particular element using predefined search method in the Collection framework. [15 M]

```

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

public class SetB1collection
{
    public static void main(String[] args) throws IOException
    {
        int n;

        HashSet hs=new HashSet();

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter the no of elements");
    }
}

```

```

n=Integer.parseInt(br.readLine());
System.out.println("Enter the nos");
for(int i=0;i<n;i++)
{
int a=Integer.parseInt(br.readLine());
hs.add(a);
}
System.out.println("The elements are"+hs);
TreeSet t=new TreeSet(hs);
Iterator it=t.iterator();
System.out.println("The Sorted data");
while(it.hasNext())
System.out.println(it.next());
}
}

```

2. Write a java program to simulate traffic signal using threads. [15 M]

```

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

public class Slip6_1 extends JFrame implements Runnable {
    private int time, cnt;
    private Thread t;

    public Slip6_1() {
        setTitle("Traffic Signal Timer");
        setSize(200, 250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);

        time = 10;
    }
}

```



```
    cnt = 0;

    t = new Thread(this);

    t.start();
}

public void run() {
    try {
        while (true) {
            while (time >= 0) {
                repaint();

                Thread.sleep(50);

                time--;
            }

            cnt = (cnt + 1) % 3;

            time = (cnt == 1) ? 5 : 10;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
public void paint(Graphics g) {
    super.paint(g);

    g.setColor(Color.black);

    g.drawOval(70, 50, 50, 50);

    g.drawOval(70, 110, 50, 50);

    g.drawOval(70, 170, 50, 50);

    g.setColor(Color.black);

    g.drawString("Time : " + time, 85, 230);
}
```

```

    if (cnt == 0) {
        g.setColor(Color.green);
        g.fillOval(70, 170, 50, 50);
    } else if (cnt == 1) {
        g.setColor(Color.orange);
        g.fillOval(70, 110, 50, 50);
    } else {
        g.setColor(Color.red);
        g.fillOval(70, 50, 50, 50);
    }
}

public static void main(String[] args) {
    new Slip6_1();
}
}

```

Slip7

1. Write a java program that implements a multi-thread application that has three threads. First thread generates random integer number after every one second, if the number is even; second thread computes the square of that number and print it. If the number is odd, the third thread computes the cube of that number and print it. [15 M]

```
import java.util.Random;
```

```
class Slip7_1 extends Thread
```

```
{
```

```
    public void run()
```

```
    {
```

```
        Random random = new Random();
```

```

        for (int i = 0; i < 10; i++)
        {
            int rnum = random.nextInt(100);

            System.out.println("Random Integer generated : " + rnum);

            if((rnum%2) == 0) {
SquareThread sThread = new SquareThread(rnum);

                sThread.start();

            }
            else
            {
CubeThread cThread = new CubeThread(rnum);

                cThread.start();

            }
            try {

                Thread.sleep(1000);

            }
            catch (InterruptedException ex) {

                System.out.println(ex);

            }

        }
    }

    public static void main(String args[])
    {
        Slip7_1 rnThread = new Slip7_1();

        rnThread.start();

    }
}

```

```

class SquareThread extends Thread
{
    int n;

```

```

        SquareThread(int rn)
        {
            n = rn;
        }

        public void run()
        {
            System.out.println("Square of " + n + " = " + (n * n));
        }
    }

```

```

class CubeThread extends Thread
{
    int n;
    CubeThread(int rn)
    {
        n = rn;
    }
    public void run()
    {
        System.out.println("Cube of " + n + " = " + (n * n * n));
    }
}

```

2. Write a java program for the following:

- i. To create a Product(Pid, Pname, Price) table.
- ii. Insert at least five records into the table.
- iii. Display all the records from a table. [15 M]

```

import java.sql.*;
import java.util.*;
public class JDBC_Slip7

```

```

{
    public static void main(String[] args)
    {
        Scanner s =new Scanner(System.in);

        try
        {
            Class.forName("org.postgresql.Driver");

            Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

            System.out.println("Enter how many records you want to insert");
            int n=s.nextInt();

            for(int i=0;i<n;i++)
            {
                System.out.println("Enter the product "+(i+1)+"details");
                System.out.println("Enter Product id");

                int pno=s.nextInt();

                System.out.println("Enter Product Name");
                String name=s.next();

                System.out.println("Enter Product Price");

                int pr=s.nextInt();

                PreparedStatement ps=con.prepareStatement("insert into product
values(?,?,?)");

                ps.setInt(1,pno);

                ps.setString(2,name);

                ps.setInt(3,pr);

                ps.executeUpdate();

            }

            Statement t=con.createStatement();

            ResultSet rs=t.executeQuery("select * from product");

            System.out.println("Product Number      Product Name Product Price ");

            while(rs.next())

```

```

        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getInt(3));
    }
catch(Exception e){System.out.println(e);}
}
}

```

Slip8

1. Write a java program to define a thread for printing text on output screen for 'n' number of times. Create 3 threads and run them. Pass the text 'n' parameters to the thread constructor. Example: i. First thread prints "COVID19" 10 times. ii. Second thread prints "LOCKDOWN2020" 20 times iii. Third thread prints "VACCINATED2021" 30 times [15 M]

//Slip no 8

```

class Print extends Thread
{
    String msg;
    int n;
    public Print(int n,String msg)
    {
        this.n=n;
        this.msg=msg;
    }
    public void run()
    {
        for(int i=0;i<n;i++)
            System.out.println(msg);
    }
}

public class Slip8_1 {
    public static void main(String[] arg)throws Exception

```

```

{
    Print t1=new Print(10,"Covid19");
    t1.start();
    Print t2=new Print(20,"Lockdown 2020");
    t2.start();
    Print t3=new Print(30,"Vaccinated 2021");
    t3.start();
}
}

```

2. Write a JSP program to check whether a given number is prime or not. Display the result in red color. [15 M]

```
<h2>Prime Number Checker</h2>
```

```

<form method="post">
    Enter a number: <input type="text" name="number">
    <input type="submit" value="Check">
</form>

```

```

<%!
boolean isPrime(int num) {
    if (num <= 1) {
        return false;
    }
    for (int i = 2; i <= Math.sqrt(num); i++) {
        if (num % i == 0) {
            return false;
        }
    }
}

```

```

        return true;
    }
    %>

<%
if (request.getMethod().equals("POST")) {
    int number = Integer.parseInt(request.getParameter("number"));
    boolean isNumberPrime = isPrime(number);
    %>
    <h2 style="color: red;">
        <%= number %> is <%= isNumberPrime ? "a Prime Number" : "Not a Prime Number" %>
    </h2>
    <% } %>

```

Slip9

1. Write a Java program to create a thread for moving a ball inside a panel vertically. The ball should be created when the user clicks on the start button. [15 M]

```

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

public class MovingBall extends JFrame {
    private int y = 0;
    private boolean moving = false;

    public MovingBall() {
        setTitle("Moving Ball");
        setSize(400, 400);
    }

```



```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
setLayout(new BorderLayout());
```

```
final BallPanel ballPanel = new BallPanel(); // Declare final
```

```
JButton startButton = new JButton("Start");
```

```
startButton.addActionListener(new ActionListener() {
```

```
    @Override
```

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

```
        BallThread thread = new BallThread(ballPanel);
```

```
        thread.start();
```

```
    }
```

```
});
```

```
add(ballPanel, BorderLayout.CENTER);
```

```
add(startButton, BorderLayout.SOUTH);
```

```
setVisible(true);
```

```
}
```

```
class BallPanel extends JPanel {
```

```
    @Override
```

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

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

```
        g.setColor(Color.RED);
```

```
        g.fillOval(180, y, 40, 40);
```

```
    }
```

```
}
```

```
class BallThread extends Thread {
```

```
    private BallPanel panel;
```

```

public BallThread(BallPanel panel) {
    this.panel = panel;
}

@Override
public void run() {
    moving = true;
    while (moving) {
        y += 5;
        if (y > panel.getHeight()) {
            y = 0;
        }
        panel.repaint();
        try {
            Thread.sleep(50);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
}

```

```

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

```

2. Write a Java program using Spring to display the message “If you can't explain it simply, you don't understand it well enough”. [15 M]

```

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;

```

```

import org.springframework.stereotype.Component;

@Component
class MessageService {

    public String getMessage() {

        return "If you can't explain it simply, you don't understand it well enough.";

    }

}

public class Slip9_2 {

    public static void main(String[] args) {

        ApplicationContext context = new
AnnotationConfigApplicationContext(MessageService.class);

        MessageService messageService = context.getBean(MessageService.class);

        System.out.println(messageService.getMessage());

    }

}

```

Slip10

1. Write a Java program to display the Current Date using spring. [15 M]

```

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.stereotype.Component;

import java.time.LocalDate;

@Component
class DateService {

    public String getCurrentDate() {

```

```

        return "Current Date: " + LocalDate.now();
    }
}

public class SpringDateApp {
    public static void main(String[] args) {
        ApplicationContext context = new AnnotationConfigApplicationContext(DateService.class);
        DateService dateService = context.getBean(DateService.class);
        System.out.println(dateService.getCurrentDate());
    }
}

```

2. Write a Java program to display first record from student table (RNo, SName, Per) onto the TextFields by clicking on button. (Assume Student table is already created). [15 M]

```

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

class Slip10_2 extends JFrame implements ActionListener
{
    JLabel l1,l2,l3;
    JTextField t1,t2,t3;
    JButton btn;
    Connection con=null;
    ResultSet rs=null;
    Slip10_2()
    {
        l1= new JLabel("Roll No:");
        l2= new JLabel("Student Name:");
        l3= new JLabel("Percentage:");
        t1= new JTextField();

```

```

t2= new JTextField();
t3= new JTextField();
btn = new JButton("Show");
    setLayout(new FlowLayout());
setTitle("Employee Information");
setSize(250,200);

    Panel p1=new Panel();
p1.setLayout(new GridLayout(3,2,20,20));
p1.add(l1);p1.add(t1);
    p1.add(l2);p1.add(t2);
    p1.add(l3);p1.add(t3);
    add(p1);
add(btn);
try{
    Class.forName("org.postgresql.Driver");
    con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");
    }catch(Exception e)
    {
        System.out.println(e);
    }
    btn.addActionListener(this);
    setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent ae)
{
    try{
        Statement st=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
        ResultSet rs = st.executeQuery("SELECT * FROM student");

```

```

        rs.first();

        t1.setText(""+rs.getInt(1));

        t2.setText(""+rs.getString(2));

        t3.setText(""+rs.getInt(3));

    }catch(Exception e){}

    }

    public static void main(String arg[])

    {

        new Slip10_2();

    }

}

```

Slip11

1. Design an HTML page which passes customer number to a search servlet. The servlet searches for the customer number in a database (customer table) and returns customer details if found the number otherwise display error message. [15 M]

//customer.html

```

<html>

<head>

    <title>Customer Search</title>

</head>

<body>

    <h2>Customer Search</h2>

    <form action="http://localhost:8080/examples/Customer" method="post">

        Enter Customer Number:<input type="text" name="customerNumber"><br><br>

        <input type="submit" value="Search">

    </form>

</body>

</html>

```

```

//Customer.java

import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.sql.*;

public class Customer extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        int cno = Integer.parseInt(request.getParameter("customerNumber"));

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        try{

            Class.forName("org.postgresql.Driver");

            Connection c =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

            PreparedStatement ps =c.prepareStatement

                ("select * from customer where id=?");

            ps.setInt(1, cno);

            ResultSet rs =ps.executeQuery();

            if(rs.next())

            {

                out.println("Customer Details");

                out.println("Customer Id = "+rs.getInt(1)+"<br><br>");

                out.println("Customer Name = "+rs.getString(2));

            }

            else

            {

                out.println("Customer Not Found");

                response.sendRedirect("Customer.html");

            }

        }

    }

}

```

```

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

    }
}

```

2. Write a Java program to display information about all columns in the DONAR table using ResultSetMetaData. [15 M]

```

import java.sql.*;

public class ResultSetMetaDataExample
{
    public static void main(java.lang.String[] args)
    {
        try {
            // Get a database connection and prepare a statement.
            Class.forName("org.postgresql.Driver");

            Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

            Statement s = con.createStatement();

            ResultSet rs = s.executeQuery("SELECT * FROM DONAR");

            ResultSetMetaData rsmd = rs.getMetaData();

            int colCount = rsmd.getColumnCount();

            int rowCount = 0;

            for (int i = 1; i <= colCount; i++)
            {
                System.out.println("Information about column " + i);

                System.out.println("  Name.....: " + rsmd.getColumnName(i));

                System.out.println("  Data Type.....: " + rsmd.getColumnType(i) +
                    " ( " + rsmd.getColumnTypeName(i) + " )");
            }
        }
    }
}

```



```

        System.out.print (" Allows Nulls..: ");
        if (rsmd.isNullable(i)==0)
            System.out.println("false");
        else
            System.out.println("true");
    }

} catch (Exception e) {
    // Handle any errors.
    System.out.println("Oops... we have an error... ");
    e.printStackTrace();
}
}
}
}

```

Slip12

1. Write a JSP program to check whether given number is Perfect or not. (Use Include directive).
[15 M]

```
<h2>Perfect Number Checker</h2>
```

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

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

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

```
</form>
```

```
<%!
```

```

boolean isPerfectNumber(int num) {
    if (num <= 0) {
        return false;
    }

    int sum = 0;
    for (int i = 1; i <= num / 2; i++) {
        if (num % i == 0) {
            sum += i;
        }
    }

    return sum == num;
}
%>

<%
if (request.getMethod().equals("POST")) {
    int number = Integer.parseInt(request.getParameter("number"));
    boolean isNumberPerfect = isPerfectNumber(number);
%>

<p>
    The number <%= number %> is

    <%= isNumberPerfect ? "perfect" : "not perfect" %>

</p>
<% } %>

```

2. Write a Java Program to create a PROJECT table with field's project_id, Project_name, Project_description, Project_Status. Insert values in the table. Display all the details of the PROJECT table in a tabular format on the screen.(using swing). [15 M]

```
import javax.swing.*;

import java.sql.*;

public class TableExample extends JFrame
{
    public TableExample()
    {
        try
        {
            int cnt=0;

            Class.forName("org.postgresql.Driver");

            Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

            String query = "SELECT * FROM project";

            Statement st = con.createStatement();

            ResultSet rs = st.executeQuery( query );

            ResultSetMetaData rsm=rs.getMetaData();

            cnt=rsm.getColumnCount();

            Object[][] data=new Object[cnt][4];

            String[] columns = new String[] {"Project Id", "Project Name", "Project
Description","Project Status"};

            for(int i=0;i<cnt & rs.next();i++)
            {
                data[i]=new Object[4];

                data[i][0]=new Integer(rs.getInt(1));

                data[i][1]=(Object)rs.getString(2);

                data[i][2]=(Object)rs.getString(3);

                data[i][3]=(Object)rs.getString(4);

            }

            //create table with data
```

```

        JTable table = new JTable(data, columns);

        //add the table to the frame

        this.add(new JScrollPane(table));
    }
catch(Exception e){System.out.println(e);}

    this.setTitle("Table Example");

    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    this.setSize(700,500);

    this.setVisible(true);
}

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

```

Slip13

1. Write a Java program to display information about the database and list all the tables in the database. (Use DatabaseMetaData). [15 M]

```

import java.sql.*;

import java.io.*;

class DBMetadata
{
    public static void main(String arg[])
    {
        try
        {

```

```

Class.forName("org.postgresql.Driver");

    Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

DatabaseMetaData md = con.getMetaData();

System.out.println(" DatabaseMetaData Information");

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

System.out.println(" - Product Name      : "+md.getDatabaseProductName());
System.out.println(" - Driver version Name      : "+md.getDriverVersion());
System.out.println(" - Product Version Number : "+md.getDatabaseProductVersion());
//System.out.println(" - Database Major Version : "+md.getDatabaseMajorVersion());
System.out.println(" - Driver Name      : "+md.getDriverName());
System.out.println(" - Username      : "+md.getUserName());
System.out.println(" - Driver Minor Version : " + md.getDriverMinorVersion());
ResultSet resultSet = md.getColumns(null, null, "student", null);

    while (resultSet.next()) {

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

        String type = resultSet.getString("TYPE_NAME");

        int size = resultSet.getInt("COLUMN_SIZE");


        System.out.println("Column name: [" + name + "]; type: [" + type
            + "]; size: [" + size + "]);

    }

ResultSet resultset1 =md.getTables(null,null,null,new String[]{"TABLE"});
while(resultset1.next())
{
String tname=resultset1.getString("TABLE_NAME");
System.out.println(" Table "+tname);

}

}

```

```

catch(Exception e)
{
System.out.println("Error : "+e);
}
}
}

```

2. Write a Java program to show lifecycle (creation, sleep, and dead) of a thread. Program should print randomly the name of thread and value of sleep time. The name of the thread should be hard coded through constructor. The sleep time of a thread will be a random integer in the range 0 to 4999. [15 M]

```

import java.util.Random;

```

```

public class Slip13_2 implements Runnable
{
    private String t_name;
    private int sleep_Time;
    public Slip13_2(String name)
    {
        t_name = name;
        Random rand = new Random();
        sleep_Time = rand.nextInt(5000);
    }
    public void run()
    {
        System.out.println(t_name + " created.\n");
        try
        {
            Thread.sleep(sleep_Time);
            System.out.println(t_name + "sleeps for " + sleep_Time +" milliseconds.\n");
        }
        catch (Exception e)

```

```

        {
            System.out.println("Exception :"+e);
        }

        System.out.println(t_name + " dead.");
    }

    public static void main(String[] args)
    {
        Thread t1 = new Thread(new Slip13_2("Thread 1..."));
        Thread t2 = new Thread(new Slip13_2("Thread 2..."));

        t1.start();
        t2.start();

        try
        {
            t1.join();
            t2.join();
        }

        catch (Exception e)
        {
            System.out.println("Exception "+e);
        }

        System.out.println("Main thread Finish successfully.");
    }
}

```

Slip14

1. Write a Java program for a simple search engine. Accept a string to be searched. Search the string in all text files in the current folder. Use a separate thread for each file. The result should display the filename and line number where the string is found. [15 M]

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

```
public class SearchThread_SetB2 extends Thread
```

```
{
```

```
    File f1;
```

```
    String fname;
```

```
    static String str;
```

```
    String line;
```

```
    LineNumberReader reader = null;
```

```
    SearchThread_SetB2(String fname)
```

```
    {
```

```
        this.fname=fname;
```

```
        f1=new File(fname);
```

```
    }
```

```
    public void run()
```

```
    {
```

```
        try
```

```
        {
```

```
            FileReader fr=new FileReader(f1);
```

```
            reader=new LineNumberReader(fr);
```

```
            while((line=reader.readLine())!=null)
```

```
            {
```

```
                if(line.indexOf(str)!=-1)
```

```
                {
```

```
                    System.out.println("string found in "+fname+"at "+reader.getLineNumber()+"line");
```

```
                    stop();
```

```
                }
```

```
            }
```

```
        }
```

```
        catch(Exception e)
```

```
        {
```



```

    }
}
public static void main(String[] args) throws IOException
{
    Thread t[]=new Thread[20];

    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

    System.out.println("Enter String to search");

    str=br.readLine();


    FilenameFilter filter = new FilenameFilter()
    {
        public boolean accept(File file, String name)
        {
            if (name.endsWith(".txt"))
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    };


    File dir1 = new File(".");
    File[] files = dir1.listFiles(filter);

    if (files.length == 0)
    {
        System.out.println("no files available with this extension");
    }
    else

```

```

{
    for(int i=0;i<files.length;i++)
    {
        for (File aFile : files)
        {
            t[i]=new SearchThread_SetB2(aFile.getName());
            t[i].start();
        }
    }
}
}
}

```

2. Write a JSP program to calculate sum of first and last digit of a given number. Display sum in Red Color with font size 18. [15 M]

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

```
    pageEncoding="UTF-8"%>
```

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

```
<form method="post" action="Slip15.jsp">
```

```
    Enter a Number: <input type="text" name="number">
```

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

```
</form>
```

```
<%-- Retrieve the input number from the request --%>
```

```
<%
```

```
    String numberStr = request.getParameter("number");
```

```
    if(numberStr != null && !numberStr.isEmpty()) {
```

```
        // Convert the input string to an integer
```

```
        int number = Integer.parseInt(numberStr);
```

```
        // Calculate the sum of the first and last digits
```

```

int lastDigit = number % 10;

int firstDigit = 0;

while (number != 0) {
    firstDigit = number % 10;
    number = number / 10;
}

int sum = firstDigit + lastDigit;

%>

```

```

<p>Number: <%= numberStr %></p>

<p>First Digit: <%= firstDigit %></p>

<p>Last Digit: <%= lastDigit %></p>

<p style="color: red; font-size: 18px;">Sum of First and Last Digit: <%= sum %></p>

<% } %>

```

Slip 15

1. Write a java program to display name and priority of a Thread. [15 M]

```

class ThreadName_Priority extends Thread {

    public void run() {

        System.out.println("Thread Name: " + getName());

        System.out.println("Thread Priority: " + getPriority());

    }

    public static void main(String[] args) {

        ThreadName_Priority t1 = new ThreadName_Priority();

        ThreadName_Priority t2 = new ThreadName_Priority();
    }
}

```

```

        t1.setName("FirstThread");

        t2.setName("SecondThread");

        t1.setPriority(Thread.MIN_PRIORITY);

        t2.setPriority(Thread.MAX_PRIORITY);

        t1.start();

        t2.start();

    }

}

```

2. Write a SERVLET program which counts how many times a user has visited a web page. If user is visiting the page for the first time, display a welcome message. If the user is revisiting the page, display the number of times visited. (Use Cookie) [15 M]

```

import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

public class Slip15 extends HttpServlet

{

    public void doGet(HttpServletRequest req,

                        HttpServletResponse res)

    {

        PrintWriter pw=null;

        try

        {

            res.setContentType("text/html");

            pw = res.getWriter();

            Cookie []c = req.getCookies();

            int i;

            if(c==null)

            {

                pw.println("<B>Welcome");

                Cookie newCookie = new Cookie("Count","1");

                res.addCookie(newCookie);

            }

        }

    }

}

```

```

else
{
for(i=0;i<c.length;i++)
{
String s=c[i].getName();
if(s.equals("Count"))
{
String s1=c[i].getValue();
int n=Integer.parseInt(s1)+1;
pw.println("page is visited for "+n+" Times");
Cookie c2=new Cookie("Count",Integer.toString(n));
res.addCookie(c2);
break;
}
}
}
}catch(Exception e){pw.println("Error"+e);}
}
}

```

Slip16

1. Write a java program to create a TreeSet, add some colors (String) and print out the content of TreeSet in ascending order. [15 M]

```

import java.util.*;

public class TreesetDemoSetA3
{

```

```

public static void main(String[] args)
{
    TreeSet<String> ts=new TreeSet<String>();
    ts.add("blue");
    ts.add("red");
    ts.add("green");
    ts.add("blue");
    ts.add("yellow");
    ts.add("cyan");

    System.out.println("The contents of set are :"+ts);
}
}

```

2. Write a Java program to accept the details of Teacher (TNo, TName, Subject). Insert at least 5 Records into Teacher Table and display the details of Teacher who is teaching “JAVA” Subject. (Use PreparedStatement Interface) [15 M]

```

import java.sql.*;
import java.util.*;

public class Slip16_2
{
    public static void main(String[] args)
    {
        Scanner s =new Scanner(System.in);

        try
        {
            Class.forName("org.postgresql.Driver");

            Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

            System.out.println("Enter how many records you want to insert");

            int n=s.nextInt();

            for(int i=0;i<n;i++)
            {

```

```

        System.out.println("Enter the Teacher "+(i+1)+"details");

        System.out.println("Enter Teacher id");

        int tno=s.nextInt();

        System.out.println("Enter Teacher Name");

        String name=s.next();

        System.out.println("Enter Teacher Subject");

        String subject=s.next();

        PreparedStatement ps=con.prepareStatement("insert into teacher
values(?,?,?)");

        ps.setInt(1,tno);

        ps.setString(2,name);

        ps.setString(3,subject);

        ps.executeUpdate();

    }

    Statement t=con.createStatement();

    ResultSet rs=t.executeQuery("select * from teacher where subject='JAVA'");

    System.out.println("Teacher Number      Teacher Name Subject ");

    while(rs.next())

        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3));

    }

    catch(Exception e){System.out.println(e);}

    }

}

```

1. Write a java program to accept 'N' integers from a user. Store and display integers in sorted order having proper collection class. The collection should not accept duplicate elements. [15 M]

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

public class SetB1collection
{
    public static void main(String[] args) throws IOException
    {
        int n;
        HashSet hs=new HashSet();
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter the no of elements");
        n=Integer.parseInt(br.readLine());
        System.out.println("Enter the nos");
        for(int i=0;i<n;i++)
        {
            int a=Integer.parseInt(br.readLine());
            hs.add(a);
        }
        System.out.println("The elements are"+hs);
        TreeSet t=new TreeSet(hs);
        Iterator it=t.iterator();
        System.out.println("The Sorted data");
        while(it.hasNext())
        System.out.println(it.next());
    }
}
```

2. Write a Multithreading program in java to display the number's between 1 to 100 continuously in a TextField by clicking on button. (Use Runnable Interface). [15 M]

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



```

import java.awt.*;
import java.awt.event.*;

public class Slip17_2 extends JFrame implements ActionListener
{
    JButton b1,b2;
    JTextField t1;
    Container c;
    Slip17_2()
    {
        setVisible(true);
        setSize(800,800);
        c=getContentPane();
        setLayout(null);
        t1=new JTextField(500);
        c.add(t1);
        t1.setBounds(10,10,1000,30);
        b1=new JButton("start");
        c.add(b1);
        b1.setBounds(20,50,100,50);
        b1.addActionListener(this);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1)
        {
            new Mythread();
        }
    }
    class Mythread extends Thread
    {

```

```

Mythread()
{
start();
}
public void run()
{
    for(int i=1;i<=100;i++)
    {
        try        {
            Thread.sleep(1000);
        }
        catch (InterruptedException e) {
        }
        t1.setText(t1.getText()+""+i+"\n");
    }
}
}
}
public static void main(String arg[])
{
    new Slip17_2().show();
}
}

```

Slip18

1. Write a java program to display name and priority of a Thread. [15 M]

```

class ThreadName_Priority extends Thread {
    public void run() {

```

```

        System.out.println("Thread Name: " + getName());
        System.out.println("Thread Priority: " + getPriority());
    }
    public static void main(String[] args) {
        ThreadName_Priority t1 = new ThreadName_Priority();
        ThreadName_Priority t2 = new ThreadName_Priority();
        t1.setName("FirstThread");
        t2.setName("SecondThread");
        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.MAX_PRIORITY);
        t1.start();
        t2.start();
    }
}

```

2. Write a SERVLET program in java to accept details of student (SeatNo, Stud_Name, Class, Total_Marks). Calculate percentage and grade obtained and display details on page. [15 M]

//Student.html

```

<body>

    <h2>Enter Student Details</h2>

    <form action="http://localhost:8080/examples/Student" method="post">
Seat No:<input type="text" name="no"><br><br>
Student Name:<input type="text" name="name"><br><br>
Student Class:<input type="text" name="class"><br><br>
Student Total Marks:<input type="text" name="mark"><br><br>
        <input type="submit" value="Search">
    </form>
</body>
</html>

//
import java.io.*;

```

```

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

public class Student extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
    response.setContentType("text/html");

        PrintWriter out = response.getWriter();

    String seatNo = request.getParameter("no");
    String studName = request.getParameter("name");
    String studentClass = request.getParameter("class");
    int totalMarks = Integer.parseInt(request.getParameter("mark"));
    String grade="";
    double percentage = (totalMarks / 600.0) * 100;
    if (percentage >= 90)
        grade="A+";
    else if (percentage >= 80)
        grade= "A";
    else if (percentage >= 70)
        grade= "B";
    else if (percentage >= 60)
        grade= "C";
    else
        grade="D";
    out.println("<html><body>");
    out.println("<h1>Student Details</h1>");
    out.println("<p>Seat No: " + seatNo + "</p>");
    out.println("<p>Name: " + studName + "</p>");
    out.println("<p>Class: " + studentClass + "</p>");
}
}

```

```

out.println("<p>Total Marks: " + totalMarks + "</p>");
out.println("<p>Percentage: " + percentage + "%</p>");
out.println("<p>Grade: " + grade + "</p>");
out.println("</body></html>");
}
}

```

Slip19

1. Write a java program to accept 'N' Integers from a user store them into LinkedList Collection and display only negative integers. [15 M]

```

import java.util.*;

public class Slip19_1{

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        LinkedList<Integer> numbers = new LinkedList<>();

        System.out.print("Enter the number of elements: ");

        int n = scanner.nextInt();

        System.out.println("Enter " + n + " integers:");

        for (int i = 0; i < n; i++)

            numbers.add(scanner.nextInt());

        System.out.println("Negative numbers:");

        for (int num : numbers)

            if (num < 0)

                System.out.print(num + " ");

        scanner.close();

    }
}

```

```
}
```

2. Write a SERVLET application to accept username and password, search them into database, if found then display appropriate message on the browser otherwise display error message. [15 M]

```
//Login.html
```

```
<form method='post' action='http://localhost:8080/examples/Login'>
```

```
Name:<input type="text" name="username"/><br/><br/>
```

```
Password:<input type="password" name="userpass"/><br/><br/>
```

```
<input type="submit" value="login"/>
```

```
</form>
```

```
//Login.java
```

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

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

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

```
public class Login extends HttpServlet
```

```
{
```

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

```
throws ServletException, IOException
```

```
{
```

```
response.setContentType("text/html");
```

```
PrintWriter out = response.getWriter();
```

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

```
String pass = request.getParameter("userpass");
```

```
try{
```

```
Class.forName("org.postgresql.Driver");
```

```
Connection c = DriverManager.getConnection("jdbc:postgresql
```

```
://localhost:5432/testdb","postgres", "123");
```

```

PreparedStatement ps =c.prepareStatement
    ("select * from login where uname=? and pass=?");

ps.setString(1, email);
ps.setString(2, pass);
ResultSet rs =ps.executeQuery();
if(rs.next())
{
    out.println("Welcome");
}
else
{
    out.println("Username or Password incorrect");
    response.sendRedirect("login.html");
}

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

```

Slip20

1. Create a JSP page to accept a number from a user and display it in words: Example: 123 – One Two Three. The output should be in red color. [15 M]

```

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

<h1>Convert Number to Words</h1>

```

```

<form method="post" action="Slip20_1.jsp">
    Enter a Number: <input type="text" name="number">
    <input type="submit" value="Convert">
</form>

<%-- Retrieve the input number from the request --%>
<%
    String numberStr = request.getParameter("number");
    if(numberStr != null && !numberStr.isEmpty()) {
        // Array to store the word representations of digits
        String[] words = {"Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};

        // Convert the input string to an integer
        int number = Integer.parseInt(numberStr);

        // Convert the number to words
        String numberInWords = "";
        while (number > 0) {
            int digit = number % 10;
            numberInWords = words[digit] + " " + numberInWords;
            number = number / 10;
        }
    }
%>

<p>Number: <%= numberStr %></p>
<p style="color: red;">Number in Words: <%= numberInWords.trim() %></p>

<% } %>

```

2. Write a java program to blink image on the JFrame continuously. [15 M]

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



```
import java.awt.Image;

import javax.swing.ImageIcon;

import javax.swing.*;

public class Slip20_2 extends JFrame implements Runnable
{
    private JLabel L1;

    private boolean isVisible = false;

    public Slip20_2()
    {
        setLayout(new FlowLayout());

        ImageIcon ic = new ImageIcon("flower.jpg");

        L1 = new JLabel(ic);

        add(L1);

        setTitle("Blink Image");

        setSize(500,500);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setVisible(true);

        Thread t1 = new Thread(this);

        t1.start();
    }

    public void run()
    {
        try
        {
            while (true)
            {
                if (isVisible)
                {
                    L1.setVisible(false);
                }

                else
```

```

        {
            L1.setVisible(true);
        }
        isVisible = !isVisible;
        Thread.sleep(500);
    }
} catch (Exception e)
{
    System.out.println("Exception "+e);
}
}

public static void main(String[] args)
{
    new Slip20_2();
}
}

```

Slip21

1. Write a java program to accept 'N' Subject Names from a user store them into LinkedList Collection and Display them by using Iterator interface. [15 M]

```

import java.util.*;

class LinkedListSetA2
{
    public static void main(String args[])
    {
        Scanner s =new Scanner(System.in);

        System.out.println("Enter how many Names you want to enter");

        int n=s.nextInt();

        LinkedList ll = new LinkedList();

        for(int i=0;i<n;i++)

```

```

{
System.out.println("Enter Name");
ll.add(s.next());
}
System.out.println("Original contents of ll: " + ll);
}
}

```

2. Write a java program to solve producer consumer problem in which a producer produces a value and consumer consume the value before producer generate the next value. (Hint: use thread synchronization) [15 M]

```

class Q
{
    int n;

    boolean valueSet = false;

    synchronized int get()
    {
        if(!valueSet)
        try{
            wait();
        } catch(InterruptedException e)
        {
            System.out.println("InterruptedException caught");
        }

        System.out.println("Got: " + n);
        valueSet = false;
        notify();
        return n;
    }

    synchronized void put(int n)
    {

```

```

        if(valueSet)
        try{
            wait();
        } catch(InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        notify();
    }
}

```

class Producer implements Runnable

```

{
    Q q;
    Producer(Q q)
    {
        this.q = q;
        new Thread(this, "Producer").start();
    }
}

```

```

public void run() {
    int i = 0;
    while(true) {
        q.put(i++);
    }
}
}

```

class Consumer implements Runnable {

```

Q q;

Consumer(Q q) {
    this.q = q;
    new Thread(this, "Consumer").start();
}

public void run() {
    while(true) {
        q.get();
    }
}
}

public class PCFixed
{

    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

```

Slip22

1. Write a Menu Driven program in Java for the following: Assume Employee table with attributes (ENo, EName, Salary) is already created. 1. Insert 2. Update 3. Display 4. Exit. [15 M]

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

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

```

public class Slip22_1
{
    public static void main(String[] args)
    {
        Connection con;
        ResultSet rs;
        Statement t;
        PreparedStatement ps;
        Scanner s =new Scanner(System.in);
        try
        {
            Class.forName("org.postgresql.Driver");
            con = DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb",
            "postgres", "123");
            do
            {
                System.out.println("\n1.Insert\n2.Update\n3.Display\n4.Exit");
                System.out.println("Enter the choice");
                int ch=s.nextInt();
                switch(ch)
                {
                    case 1:
                        System.out.println("Enter Employee_Number");
                        int eno=s.nextInt();
                        System.out.println("Enter Employee Name");
                        String ename=s.next();
                        System.out.println("Enter Salary");
                        int sal=s.nextInt();
                        ps=con.prepareStatement("insert into emp values(?,?,?)");
                        ps.setInt(1,eno);
                        ps.setString(2,ename);

```

```

        ps.setInt(3,sal);

        ps.executeUpdate();

    break;

case 2:

    System.out.println("Enter the employee no to update record");

    int eno1=s.nextInt();

    System.out.println("Enter Employee Name");

    String ename1=s.next();

        System.out.println("Enter Salary");

    int Sal1=s.nextInt();

        ps=con.prepareStatement("update emp set ename=?, salary=? where empid=? ");

        ps.setInt(3,eno1);

            ps.setString(1,ename1);

            ps.setInt(2,Sal1);

            ps.executeUpdate();

        break;

case 3:

    t=con.createStatement();

    rs=t.executeQuery("select * from emp");

        System.out.println("Employee Number\tEmployee Name\t Salar");

while(rs.next())

    {

        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getInt(3));

    }

    break;

case 4:

    System.exit(0);

    break;

}

}while(true);

    }

```

```
catch(Exception e){System.out.println(e);}

}

}
```

2. Write a JSP program which accepts UserName in a TextBox and greets the user according to the time on server machine. [15 M]

//name.html

```
<form method='post' action='http://localhost:8080/user.jsp'>
```

```
User Name:<input type='text' name='uname'><br>
```

```
<input type='submit'><input type='reset'>
```

```
</form>
```

//greet.jsp

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

```
<%
```

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

```
    Date d = new Date();
```

```
    if(d.getHours()<12)
```

```
    {
```

```
%>
```

```
Good Morning <%=name%>
```

```
<%
```

```
    }
```

```
    else if(d.getHours()<16)
```

```
    {
```

```
%>
```

```
Good Afternoon <%=name%>
```

```
<%
```

```
    }
```

```
    else
```

```
    {
```

```
%>
```

```
Good Evening <%=name%>
```



```
<%  
    }  
%>
```

Slip23

1. Write a java program to accept a String from a user and display each vowel from a String after every 3 seconds. [15 M]

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

```
public class Slip23_1 extends Thread
```

```
{
```

```
    String S1;
```

```
    public Slip23_1(String str)
```

```
    {
```

```
        S1 = str;
```

```
        start();
```

```
    }
```

```
    public void run()
```

```
    {
```

```
        for (int i = 0; i < S1.length(); i++)
```

```
        {
```

```
            char ch = S1.charAt(i);
```

```
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch  
== 'u' || ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
```

```
            {
```

```
                System.out.println(ch);
```

```
            }  
        }
```

```

        {
            Thread.sleep(3000);
        }
        catch (Exception e)
        {
            System.out.println("Exception : "+e);
        }
    }
}
}

public static void main(String[] args)
{
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a string: ");
    String str2 = scanner.nextLine();
    Slip23_1 obj = new Slip23_1(str2);
}
}

```

2. Write a java program to accept 'N' student names through command line, store them into the appropriate Collection and display them by using Iterator and ListIterator interface. [15 M]

```

import java.util.*;

public class Slip23_2{
    public static void main(String[] args) {
        if (args.length == 0) {
            System.out.println("Please provide student names as command-line arguments.");
            return;
        }
        List studentNames = new ArrayList(Arrays.asList(args));
        for(int i=0;i<args.length;i++)
            studentNames.add(args[i]);
    }
}

```

```

// Display student names using Iterator

System.out.println("\nDisplaying student names using Iterator:");

Iterator<String> it= studentNames.iterator();

while (it.hasNext())

    System.out.println(it.next());

System.out.println("Student names using ListIterator (forward direction):");

ListIterator lt = studentNames.listIterator();

while (lt.hasNext())

    System.out.println(lt.next());

System.out.println("\nStudent names using ListIterator (backward direction):");

while (lt.hasPrevious())

    System.out.println(lt.previous());

}

}

```

Slip24

1. Write a java program to scroll the text from left to right continuously. [15 M]

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

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

```

public class Slip24_1 extends JPanel implements Runnable {

    private String text = " Welcome to Java ";

    private int x = 0;

    public Slip24_1() {

        new Thread(this).start();

    }
}

```

```

public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setFont(new Font("Arial", Font.BOLD, 24));
    g.drawString(text, x, 50);
}

```

```

public void run() {
    while (true) {
        x += 5;
        if (x > getWidth()) x = -text.length() * 10;
        repaint();
        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

```

public static void main(String[] args) {
    JFrame frame = new JFrame("Scrolling Text");
    Slip24_1 obj= new Slip24_1();
    frame.add(obj);
    frame.setSize(400, 100);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
}
}

```

2. Write a JSP script to accept username and password from user, if they are same then display “Login Successfully” message in Login.html file, otherwise display “Login Failed” Message in Error.html file. [15 M]

//Login.html

```

<form method='post' action='http://localhost:8080/Slip24/Validation.jsp'>
User Name:<input type='text' name='uname'><br>
Password:<input type='text' name='pass'><br>
<input type='submit'><input type='reset'>
</form>

//validation.jsp
<%@ page language="java" %>
<%
String username = request.getParameter("uname");
String password = request.getParameter("pass");
if(username.equals(password))
    response.sendRedirect("success.html");
else
    response.sendRedirect("Error.html");
%>

//success.html
<html> <head> <title>Login Successfully</title> </head> <body> <h1>Login Successfully</h1>
</body> </html>

//error.html
<html> <head> <title>Login Failed</title> </head> <body> <h1>Login Failed</h1> </body>
</html>

```

Slip25

1. Write a JSP program to accept Name and Age of Voter and check whether he is eligible for voting or not. [15 M]

// age.html

```

<form method='post' action='http://localhost:8080/Slip25/Voter.jsp'>

```

```

User Name:<input type='text' name='uname'><br>
Age:<input type='text' name='age'><br>
<input type='submit'><input type='reset'>
</form>
//checkage.jsp
<%
    String name = request.getParameter("uname");
    int age=Integer.parseInt(request.getParameter("age"));
    if(age>18)
        out.println(name+" is eligible for voting");
else
    out.println(name+" is not eligible for voting");
%>

```

2. Write a Java Program for the following: Assume database is already created. [15 M]

The image shows a Java Swing window with a title bar. Inside the window, there is a label that says "Type Your DDL Query Here". To the right of this label is a rectangular text input field. Below the input field, there are three buttons arranged horizontally: "Create Table", "Alter Table", and "Drop Table".

```

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

public class Slip25_2 extends JFrame implements ActionListener
{
    JLabel l1;

```

```

JButton create,alter,drop;

JTextField t1;

PreparedStatement ps;

Connection con;

ResultSet rs;

public Slip25_2()
{
    l1=new JLabel("Type your DDL query here:");
    create=new JButton("Create Table");
    alter=new JButton("Alter Table");
    drop=new JButton("Drop Table");
    t1=new JTextField(30);
    setLayout(new FlowLayout());
    add(l1);add(t1);
    add(create);add(alter);add(drop);
    create.addActionListener(this);
    alter.addActionListener(this);
    drop.addActionListener(this);
    try
    {
        Class.forName("org.postgresql.Driver");
        con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");
    }
catch(Exception e){System.out.println(e);}

    this.setTitle("DDL Queries");
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setSize(340,350);
    this.setVisible(true);
}

    public void actionPerformed(ActionEvent ae)

```

```

        {
            String qr=t1.getText();
            try{
Statement st=con.createStatement();
                if(ae.getSource()==create)
                {
                    boolean r=st.execute(qr);
                    System.out.println("Table Created Successfully");
                }
                if(ae.getSource()==alter)
                {
                    boolean r=st.execute(qr);
                    System.out.println("Table Altered Successfully");
                }
                if(ae.getSource()==drop)
                {
                    boolean r=st.execute(qr);
                    System.out.println("Table Dropped Successfully");
                }
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }

public static void main(String[] args)
{
    new Slip25_2();
}
}

```


Slip26

1. Write a Java program to delete the details of given employee (ENo EName Salary). Accept employee ID through command line. (Use PreparedStatement Interface) [15 M]

```
import java.sql.*;
import java.util.*;
public class Slip26_1
{
    public static void main(String[] args)
    {
        try
        {
            Class.forName("org.postgresql.Driver");
            Connection con =
            DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb",
            "postgres", "123");
            int empno=Integer.parseInt(args[0]);
            PreparedStatement ps=con.prepareStatement("delete from emp where empid=?");
            ps.setInt(1,empno);
            ps.executeUpdate();
            System.out.println("Record deleted successfully");
        }
        catch(Exception e){System.out.println(e);}
    }
}
```

2. Write a JSP program to calculate sum of first and last digit of a given number. Display sum in Red Color with font size 18. [15 M]

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<h1>Calculate Sum of First and Last Digit</h1>

<form method="post" action="Slip15.jsp">
```

```
Enter a Number: <input type="text" name="number">
<input type="submit" value="Calculate">
</form>

<%-- Retrieve the input number from the request --%>
<%
String numberStr = request.getParameter("number");
if(numberStr != null && !numberStr.isEmpty()) {
    // Convert the input string to an integer
    int number = Integer.parseInt(numberStr);

    // Calculate the sum of the first and last digits
    int lastDigit = number % 10;
    int firstDigit = 0;
    while (number != 0) {
        firstDigit = number % 10;
        number = number / 10;
    }
    int sum = firstDigit + lastDigit;
}%>

<p>Number: <%= numberStr %></p>
<p>First Digit: <%= firstDigit %></p>
<p>Last Digit: <%= lastDigit %></p>
<p style="color: red; font-size: 18px;">Sum of First and Last Digit: <%= sum %></p>

<% } %>
```

Slip27

1. Write a Java Program to display the details of College (CID, CName, address, Year) on JTable.
[15 M]

```
import javax.swing.*;

import java.sql.*;

public class Slip27_1 extends JFrame
{
    public Slip27_1()
    {
        try
        {
            int cnt=0;

            Class.forName("org.postgresql.Driver");

            Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

            String query = "SELECT * FROM college";

            Statement st = con.createStatement();

            ResultSet rs = st.executeQuery( query );

            ResultSetMetaData rsm=rs.getMetaData();

            cnt=rsm.getColumnCount();

            Object[][] data=new Object[cnt][4];

            String[] columns = new String[] { "College Id", "College Name", "College
Address","Establishment Year" };

            for(int i=0;i<cnt & rs.next();i++)
            {
                data[i]=new Object[4];

                data[i][0]=new Integer(rs.getInt(1));

                data[i][1]=(Object)rs.getString(2);

                data[i][2]=(Object)rs.getString(3);

                data[i][3]=new Integer(rs.getInt(4));

            }
        }
    }
}
```

```

        //create table with data

        JTable table = new JTable(data, columns);

        //add the table to the frame

        this.add(new JScrollPane(table));

    }

    catch(Exception e){System.out.println(e);}

    this.setTitle("Table Example");

    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    this.setSize(700,500);

    this.setVisible(true);

}

public static void main(String[] args)

{

    new Slip27_1();

}

}

```

2. Write a SERVLET program to change inactive time interval of session. [15 M]

```

import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.sql.*;

public class Slip27 extends HttpServlet

{

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException

{

    int newTimeout = 3600;

    HttpSession session = request.getSession();

    session.setMaxInactiveInterval(newTimeout);

```

```

response.setContentType("text/html");

response.getWriter().println("Session inactive timeout has been changed to " + newTimeout +
" seconds.");
}
}

```

Slip 28

1. Write a JSP script to accept a String from a user and display it in reverse order. [15 M]

<h2>Reverse String</h2>

```

<form method="post">

    Enter a String: <input type="text" name="str">

    <input type="submit" value="Reverse">

</form>

```

```

<%
if (request.getMethod().equals("POST")) {
    String st = request.getParameter("str");
    StringBuffer sb=new StringBuffer(st);
    sb=sb.reverse();
%>
<p>
    The Reverse String is <%= sb %>
</p>
<% } %>

```

2. Write a java program to display name of currently executing Thread in multithreading. [15 M]

```

class Slip28_2 extends Thread {

```

```

public void run() {
    System.out.println("Currently executing thread: " + Thread.currentThread().getName());
}

public static void main(String[] args) {

    Slip28_2 t1 = new Slip28_2();
    Slip28_2 t2 = new Slip28_2();
    t1.setName("FirstThread");
    t2.setName("SecondThread");
    t1.start();
    t2.start();

    System.out.println("Main thread: " + Thread.currentThread().getName());
}
}

```

Slip29

1. Write a Java program to display information about all columns in the DONAR table using ResultSetMetaData. [15 M]

```

import java.sql.*;

public class ResultSetMetaDataExample
{
    public static void main(java.lang.String[] args)
    {

        Statement s = null;

```

```

ResultSet rs = null;

ResultSetMetaData rsmd = null;

try {
    // Get a database connection and prepare a statement.

    Class.forName("org.postgresql.Driver");

    Connection c = DriverManager.getConnection("jdbc:postgresql://
localhost:5432/testdb","postgres", "123");

    s = c.createStatement();

    rs = s.executeQuery("SELECT * FROM DONAR");

    rsmd = rs.getMetaData();

    int colCount = rsmd.getColumnCount();

    int rowCount = 0;

    for (int i = 1; i <= colCount; i++)
    {
        System.out.println("Information about column " + i);

        System.out.println("  Name.....: " + rsmd.
getColumnName(i));

        System.out.println("  Data Type.....: " + rsmd.
getColumnType(i) +
" ( " + rsmd.getColumnTypeName(i) + " )");

        System.out.print ("  Allows Nulls..: ");

        if (rsmd.isNullable(i)==0)

            System.out.println("false");

        else

            System.out.println("true");

    }
} catch (Exception e) {

    // Handle any errors.

    System.out.println("Oops... we have an error... "+e);

}
}

```

```
}
```

2. Write a Java program to create LinkedList of integer objects and perform the following: i. Add element at first position ii. Delete last element iii. Display the size of link list [15 M]

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

```
public class Slip29_2{
```

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

```
        // Create a LinkedList of Integer objects
```

```
        LinkedList<Integer> numbers = new LinkedList<>();
```

```
        // Adding some initial elements to the LinkedList
```

```
        numbers.add(10);
```

```
        numbers.add(20);
```

```
        numbers.add(30);
```

```
        numbers.add(40);
```

```
        // i. Add element at the first position
```

```
        numbers.addFirst(5);
```

```
        System.out.println("After adding element at first position: " + numbers);
```

```
        // ii. Delete the last element
```

```
        numbers.removeLast();
```

```
        System.out.println("After deleting the last element: " + numbers);
```

```
        // iii. Display the size of the LinkedList
```

```
        int size = numbers.size();
```

```
        System.out.println("Size of the LinkedList: " + size);
```

```
    }
```

```
}
```


Slip30

1. Write a java program for the implementation of synchronization. [15 M]

```
class Nsynchro
{
    synchronized void disp()
    {
        try
        {
            System.out.println("Come on");
            System.out.println("Students");
            System.out.println("enjoy");
            Thread.sleep(1000);
        }
        catch (InterruptedException e){System.out.println("Error"+e);}

        System.out.println("book");
        System.out.println("Reading");
    }
}

class Test implements Runnable
{
    Thread d;
    Nsynchro ob1;

    public Test(Nsynchro o1)
    {
        ob1=o1;
    }
}
```

```

d=new Thread(this);
d.start();
}
public void run()
{
ob1.disp();
}
}
class Testmain
{
public static void main(String []arg)
{
Nsynchro ob2=new Nsynchro();
Test ob3=new Test(ob2);
Test ob4=new Test(ob2);
}
}

```

2. Write a Java Program for the implementation of scrollable ResultSet. Assume Teacher table with attributes (TID, TName, Salary) is already created. [15 M]

```

import java.io.*;
import java.sql.*;
import java.util.*;
class Slip30_2
{
public static void main(String args[])
{
int ch;
Scanner s=new Scanner(System.in);
try
{
Class.forName("org.postgresql.Driver");

```

```

Connection con =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/testdb","postgres", "123");

Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);

ResultSet rs = stmt.executeQuery("select * from Teacher1");

int count=0;

while(rs.next())

count++;

System.out.println("Which Record u want");

System.out.println("Records are = "+count);

do

{ System.out.println("1 First \n2 last \n3 next \n4 prev \n0 Exit");

ch=s.nextInt();

switch(ch)

{

case 1: rs.first();

System.out.println("Teacher Id :"+rs.getInt(1)+" Name :"+rs.getString(2)+" Salary:"+rs.getInt(3));

break;

case 2: rs.last();

System.out.println("Teacher Id :"+rs.getInt(1)+" Name :"+rs.getString(2)+" Salary:"+rs.getInt(3));

break;

case 3 : rs.next();

if(rs.isAfterLast())

System.out.println("can't move forward");

else

System.out.println("Teacher Id :"+rs.getInt(1)+" Name :"+rs.getString(2)+" Salary:"+rs.getInt(3));

break;

case 4 : rs.previous();

if(rs.isBeforeFirst())

System.out.println("can't move backward");

else

System.out.println("Teacher Id :"+rs.getInt(1)+" Name :"+rs.getString(2)+" Salary:"+rs.getInt(3));

```

```
break;
case 0 : break;
default: System.out.println("Enter valid operation");
} //switch
}while(ch!=0);
} //end of try
catch(Exception e)
{
System.out.println(e);
}
} //main
} //class
```