Slip 1/\*

 Write a Java program using Multithreading to display all the alphabets between ‘A’ to

‘Z’ after every 2 seconds.

 \*/

package com.mycompany.javaslip;

import java.util.logging.\*;

public class slip1\_1

{

    public static void main(String[] args)

    {

        Thread t = new Thread(() ->

        {

            while(true)

            {

                for(char ch = 'A'; ch <= 'Z'; ch++)

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

                System.out.println();

                try

                {

                    Thread.sleep(2000);

                }

                catch (InterruptedException ex)

                {

                    Logger.getLogger(slip1\_1.class.getName()).log(Level.SEVERE, null, ex);

                }

                System.out.println("2 seconds are passed....");

            }

        });

        t.start();

    }

}

/\*

Slip no 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)

 \*/

package com.mycompany.prac1;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.io.IOException;

import java.sql.\*;

import java.util.logging.\*;

import javax.swing.\*;

class EmpApp {

    private JFrame frame;

    private JTextField eno, ename, desig, sal;

    private JButton clear, insert;

    EmpApp() throws SQLException {

        frame = new JFrame("Employee App");

        frame.setSize(400, 200);

        frame.setLayout(new GridLayout(5,2));

        eno = new JTextField();

        ename = new JTextField();

        desig = new JTextField();

        sal = new JTextField();

        frame.add(new JLabel("Eno."));

        frame.add(eno);

        frame.add(new JLabel("EName"));

        frame.add(ename);

        frame.add(new JLabel("Designation"));

        frame.add(desig);

        frame.add(new JLabel("Salary"));

        frame.add(sal);

        clear = new JButton("Clear");

        insert = new JButton("insert");

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

        insert.addActionListener((ActionEvent e) -> {

            try {

                insertEmp(conn, eno, ename, desig, sal);

            } catch (IOException | SQLException ex) {

                Logger.getLogger(EmpApp.class.getName()).log(Level.SEVERE, null, ex);

            }

        });

        clear.addActionListener((ActionEvent e) -> {

            eno.setText("");

            ename.setText("");

            desig.setText("");

            sal.setText("");

        });

        frame.add(insert);

        frame.add(clear);

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

        frame.setVisible(true);

    }

    private static void insertEmp(Connection conn, JTextField eno, JTextField ename, JTextField desig, JTextField sal)

            throws IOException, SQLException {

        String sql = "insert into emp values(?, ?, ?, ?)";

        PreparedStatement ps = conn.prepareStatement(sql);

        ps.setInt(1, Integer.parseInt(eno.getText()));

        ps.setString(2, ename.getText());

        ps.setString(3, desig.getText());

        ps.setFloat(4, Float.parseFloat(sal.getText()));

        ps.executeUpdate();

    }

}

public class slip1\_2

{

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

        new EmpApp();

    }

}

/\*

 Slip no 2

Q1 Write a java program to read ‘N’ names of your friends, store it into HashSet and

display them in ascending order.

 \*/

package com.mycompany.practical\_slip;

import java.util.\*;;

public class slip2\_1

 {

    public static void main(String[] args)

 {

        HashSet<String> friends = new HashSet<>();

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter N :");

        int n  = scan.nextInt();

        scan.nextLine();

        for(int i = 0 ; i<n;i++)

        {

            System.out.println("Enter name :");

            String name = scan.nextLine();

            friends.add(name);

        }

        TreeSet<String> tree = new TreeSet<>(friends);

        System.out.println(tree);

    }

}

/\*

 Slip no 3

Q1. Write a JSP program to display the details of Patient (PNo, PName, Address, age,

disease) in tabular form on browser\*/

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <h1>Patient</h1>

        <table border="1">

            <tr>

                <th>PNo</th>

                <th>PName</th>

                <th>Address</th>

                <th>age</th>

                <th>disease</th>

            </tr>

            <tr>

                <td>1</td>

                <td>John</td>

                <td>xyz</td>

                <td>45</td>

                <td>kovid</td>

            </<tr>

            <tr>

                <td>2</td>

                <td>Brock</td>

                <td>abc</td>

                <td>48</td>

                <td>canser</td>

            </<tr>

        </table>

    </body>

</html>

\*/

/\*

Slip no 3  Q2. 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

 \*/

package com.mycompany.javaslip;

import java.util.\*;

public class slip3\_2 {

    public static void main(String[] args) {

        LinkedList<String> names = new LinkedList<>();

        Scanner sc = new Scanner(System.in);

        int ch;

        do {

            System.out.println("Menu");

            System.out.println("1. Insert at tail");

            System.out.println("2. Delete head.");

            System.out.println("3. Display in reverse");

            System.out.println("4. Exit");

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

            System.out.println("Enter your choice:");

            ch = sc.nextInt();

            sc.nextLine();

            System.out.println();

            switch (ch) {

                case 1:

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

                    names.add(sc.nextLine());

                    break;

                case 2:

                    names.remove();

                    break;

                case 3:System.out.println("Real order");

                    Iterator itr = names.iterator();

                    while (it.hasNext())

                    {

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

                    }

                    Iterator it = names.descendingIterator();

                    while (it.hasNext())

                    {

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

                    }

                    break;

                default:

                    System.out.println("Invalid choice.");

            }

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

        } while (ch != 4);

    }

}

/\*

Slip no 4 Q1 Write a Java program using Runnable interface to blink Text on the JFrame (Use

Swing)

 \*/

package com.mycompany.practical\_slip;

import java.awt.Color;

import java.util.Random;

import javax.swing.\*;

class BlinkText implements Runnable

{

    private JFrame frame;

    private JLabel blink;

    public BlinkText() {

        frame = new JFrame("Blink Light");

        frame.setSize(200, 200);

        blink = new JLabel("Blink");

        frame.add(blink);

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

        frame.setVisible(true);

    }

    @Override

    public void run() {

        Random rand = new Random();

        while(true) {

            int r = rand.nextInt(255);

            int g = rand.nextInt(255);

            int b = rand.nextInt(255);

            blink.setForeground(new Color(r, g, b));

        }

    }

}

public class slip4\_1

{

    public static void main(String[] args) {

        Thread t = new Thread(new BlinkText());

        t.start();

    }

}

/\*

Slip no 4 Q2. 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

 \*/

package com.mycompany.practical\_slip;

import java.util.\*;

public class slip4\_2

{

    public static void main(String[] args) {

        Map<String, String> cityMap = new HashMap<>();

        Scanner sc = new Scanner(System.in);

        int ch;

        String code, city;

        do {

            System.out.println("Menu");

            System.out.println("1. Add City and std code.(no duplicates)");

            System.out.println("2. Remove City.");

            System.out.println("3. Search city name dsiplay std code");

            System.out.println("4. Exit");

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

            System.out.println("Enter your choice:");

            ch = sc.nextInt();

            sc.nextLine();

            System.out.println();

            switch(ch) {

                case 1: System.out.println("Enter std code.");

                    code = sc.nextLine();

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

                    city = sc.nextLine();

                    cityMap.put(code, city);

                    break;

                case 2: System.out.println("Enter std code.");

                    code = sc.nextLine();

                    cityMap.remove(code);

                    break;

                case 3: System.out.println("Enter city:");

                    city = sc.nextLine();

                    code = null;

                    for(Map.Entry<String, String> map : cityMap.entrySet()) {

                        if(map.getValue().equals(city))

                            code = map.getKey();

                    }

                    if(code != null)

                        System.out.println("Code is " + code);

                    else

                        System.out.println("Not found.");

                    break;

                default: System.out.println("Invalid choice.");

            }

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

        } while(ch != 4);

    }

}

/\*

Slip no5 Q1. 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

 \*/

package com.mycompany.javaslip;

import java.util.\*;

public class slip5\_1

{

    public static void main(String[] args)

    {

        Hashtable<String, String> studentTable = new Hashtable<>();

        studentTable.put("1234567890", "john");

        studentTable.put("1239874560", "carry");

        Enumeration<String> moblieNumbers = studentTable.keys();

        while(moblieNumbers.hasMoreElements())

        {

            String no = moblieNumbers.nextElement();

            String name = studentTable.get(no);

            System.out.println("Student name: " + name + ", Mobile no: " + no);

        }

    }

}

/\*

 slip no 6 Q1 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

 \*/

package com.mycompany.practical\_slip;

import java.util.\*;

public class slip6\_1

{

    public static void main(String[] args) {

        TreeSet<Integer> nums = new TreeSet<>();

        Scanner sc = new Scanner(System.in);

        System.out.println("How many number:");

        int n = sc.nextInt();

        System.out.println("Eneter " + n + " values:");

        for(int i=0; i<n; i++)

            nums.add(sc.nextInt());

        System.out.println(nums);

        System.out.println("Enter key to search:");

        int key = sc.nextInt();

        if(nums.contains(key))

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

        else

            System.out.println("Not found.");

    }

}

/\*

 slip no 6 q2 Write a java program using multithreading to simulate traffic signal (Use Swing).

\*/

package com.mycompany.practical\_slip;

import java.util.logging.\*;

class TrafficLight implements Runnable {

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

    @Override

    public void run() {

        int idx = 0;

        while(true) {

            System.out.println("Current Signal : " + lights[idx]);

            try {

                Thread.sleep(getDuration(lights[idx]) \* 1000);

            } catch (InterruptedException ex) {

                Logger.getLogger(TrafficLight.class.getName()).log(Level.SEVERE, null, ex);

            }

            idx = (idx + 1) % lights.length;

        }

    }

    private int getDuration(String light) {

        switch(light) {

            case "Red": return 4;

            case "Green": return 7;

            case "Yellow": return 2;

            default : return 0;

        }

    }

}

public class slip6\_2

{

    public static void main(String[] args) {

        Thread t = new Thread(new TrafficLight());

        t.start();

    }

}

/\*

slip no 7 Q2 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 prints it. If the number is

odd, the third thread computes the cube of that number and prints it.

 \*/

package com.mycompany.practical\_slip;

import java.util.Random;

import java.util.logging.\*;

class NumGenerator implements Runnable {

    Random rand = new Random();

    int n;

    @Override

    public void run() {

        while(true) {

            n = rand.nextInt(100);

            System.out.println("Generated number: " + n);

            try {

                Thread.sleep(1000);

            } catch (InterruptedException ex) {

                Logger.getLogger(NumGenerator.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    }

}

class SqrGenerator implements Runnable {

    NumGenerator numGenerator;

    SqrGenerator(NumGenerator numGenerator) {

        this.numGenerator = numGenerator;

    }

    @Override

    public void run() {

        while(true) {

            int n = numGenerator.n;

            if(n % 2 == 0)

                System.out.println("Square of " + n + " is " + n\*n);

            try {

                Thread.sleep(1000);

            } catch (InterruptedException ex) {

                Logger.getLogger(SqrGenerator.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    }

}

class CubeGenerator implements Runnable {

    NumGenerator numGenerator;

    int n;

    CubeGenerator(NumGenerator numGenerator) {

        this.numGenerator = numGenerator;

    }

    @Override

    public void run() {

        while(true) {

            int n = numGenerator.n;

            if(n % 2 != 0)

                System.out.println("Cube of " + n + " is " + n\*n\*n);

            try {

                Thread.sleep(1000);

            } catch (InterruptedException ex) {

                Logger.getLogger(CubeGenerator.class.getName()).log(Level.SEVERE, null, ex);

            }

        }

    }

}

public class slip7\_1

{

    public static void main(String[] args) {

        NumGenerator numGenerator = new NumGenerator();

        Thread t1 = new Thread(numGenerator);

        t1.start();

        SqrGenerator sqrGenerator = new SqrGenerator(numGenerator);

        Thread t2 = new Thread(sqrGenerator);

        t2.start();

        CubeGenerator cubeGenerator = new CubeGenerator(numGenerator);

        Thread t3 = new Thread(cubeGenerator);

        t3.start();

    }

}

/\*

 slip no 7 q2. Write a java program for the following:

i. To create a Product (Pid, Pname, Price) table.

ii. Insert at least five records into the Product table.

iii. Display all the records from a Product table.

Assume Database is already created

 \*/

package com.mycompany.practical\_slip;

import java.sql.\*;

import java.util.Scanner;

public class slip7\_2

{

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

        Scanner sc = new Scanner(System.in);

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

        int ch;

        do {

            System.out.println("Menu");

            System.out.println("1. Create table Product.");

            System.out.println("2. Insert into Product.");

            System.out.println("3. Display records of product.");

            System.out.println("4. Exit.");

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

            System.out.println("Enter your choice:");

            ch = sc.nextInt();

            switch(ch) {

                case 1: create(conn);

                    break;

                case 2: insert(conn);

                    break;

                case 3 : select(conn);

                    break;

                default : System.out.println("Invalid choice.");

                    break;

            }

        } while(ch != 4);

    }

    private static void create(Connection conn) throws SQLException {

        String sql = "create table if not exists product("

                        + "pid int primary key,"

                        + "pname varchar(30),"

                        + "price decimal(10, 2))";

        Statement stmt = conn.createStatement();

        stmt.execute(sql);

    }

    private static void insert(Connection conn) throws SQLException {

        String sql = "insert into product values(?, ?, ?)";

        PreparedStatement pt = conn.prepareStatement(sql);

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter pid:");

        int pid = sc.nextInt();

        sc.nextLine();

        System.out.println("Enter pname:");

        String name = sc.nextLine();

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

        float price = sc.nextFloat();

        pt.setInt(1, pid);

        pt.setString(2, name);

        pt.setFloat(3, price);

        pt.executeUpdate();

    }

    private static void select(Connection conn) throws SQLException {

        String sql = "select \* from product";

        Statement stmt = conn.createStatement();

        stmt.executeQuery(sql);

        ResultSet res = stmt.getResultSet();

        while(res.next()) {

            System.out.println("Pid = " + res.getInt("pid"));

            System.out.println("PName = " + res.getString("pname"));

            System.out.println("Price = " + res.getFloat("price"));

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

        }

    }

}

/\*

slip no 9 Q1. 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

 \*/

package com.mycompany.practical\_slip;

class T1 extends Thread {

    String msg;

    T1(String msg) {

        this.msg = msg;

    }

    public void run() {

        for(int i=0; i<10; i++)

            System.out.println(msg);

    }

}

class T2 extends Thread {

    String msg;

    T2(String msg) {

        this.msg = msg;

    }

    public void run() {

        for(int i=0; i<20; i++)

            System.out.println(msg);

    }

}

class T3 extends Thread {

    String msg;

    T3(String msg) {

        this.msg = msg;

    }

    public void run() {

        for(int i=0; i<30; i++)

            System.out.println(msg);

    }

}

public class slip8\_1

{

    public static void main(String[] args) {

        T1 t1 = new T1("COVID19");

        T2 t2 = new T2("LOCKDOWN2020");

        T3 t3 = new T3("VACCINATED2021");

        t1.start();

        t2.start();

        t3.start();

    }

}

/\*slip no 8 Q2\*/

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

        <style>

            .prime { color: red; }

        </style>

    </head>

    <body>

        <h1>Is prime?</h1>

        <form action="S8Q2.jsp" method="post">

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

            <input type="submit" value="is prime ?">

        </form>

        <%

            String numStr = request.getParameter("num");

            int n = 0;

            if(numStr != null && !numStr.isEmpty()) {

                n = Integer.parseInt(numStr);

                if(n > 1) {

                    boolean isPrime = true;

                    for(int i=2; i<n; i++) {

                        if(n % i == 0) {

                            isPrime = false;

                            break;

                        }

                    }

                    if(isPrime) {

        %>

                        <h3 class="prime">Prime number</h3>

        <%

                    } else {

        %>

                        <h3 class="prime">Not a prime number</h3>

        <%

                    }

                }

            }

        %>

    </body>

</html>

/\*

slip no 9 Q1. 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 (Use Swing).

 \*/

package com.mycompany.practical\_slip;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.util.logging.\*;

import javax.swing.\*;

class BallPanel extends JPanel

{

    private int yDelta = 0;

    @Override

    protected void paintComponent(Graphics g)

    {

        super.paintComponent(g);

        g.setColor(Color.red);

        g.fillOval(175, yDelta, 50, 50);

        repaint();

    }

    void setBallPos(int y) {

        this.yDelta = y;

    }

}

public class slip9\_1

{

    private Thread ballThread;

    private BallPanel ballPanel;

    private JFrame frame;

    private JButton start;

    slip9\_1()

    {

        frame = new JFrame("Ball Movement App");

        frame.setSize(400, 400);

        ballPanel = new BallPanel();

        start = new JButton("Start");

        start.addActionListener((ActionEvent e) ->

        {

            startBallMovement();

        });

        frame.setLayout(new BorderLayout());

        frame.add(ballPanel, BorderLayout.CENTER);

        frame.add(start, BorderLayout.SOUTH);

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

        frame.setVisible(true);

    }

    private void startBallMovement()

    {

        if(ballThread == null || !ballThread.isAlive())

        {

            ballThread = new Thread(() -> {

                moveBallVertically();

            });

            ballThread.start();

        }

    }

    private void moveBallVertically()

    {

        int y = 0;

        int dir = 1;

        while(true)

        {

            try

            {

                Thread.sleep(15);

            } catch (InterruptedException ex)

            {

                Logger.getLogger(slip9\_1.class.getName()).log(Level.SEVERE, null, ex);

            }

            y += 5 \* dir;

            if(y > ballPanel.getHeight() - 50)

                dir = -1;

            if(y <= 0)

                dir = 1;

            ballPanel.setBallPos(y);

        }

    }

    public static void main(String[] args)

    {

        new slip9\_1();

    }

}

/\*

slip no 10 Q2. 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)

 \*/

package com.mycompany.javaslip;

import java.awt.GridLayout;

import java.sql.\*;

import java.util.logging.\*;

import javax.swing.\*;

class StudentRec

{

    private JFrame frame;

    private JTextField tf1, tf2, tf3;

    private JButton display;

    StudentRec() throws SQLException {

        frame = new JFrame("Student First Record.");

        frame.setSize(200, 300);

        tf1 = new JTextField();

        tf2 = new JTextField();

        tf3 = new JTextField();

        display = new JButton("Show Record");

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

        display.addActionListener((ActionEvent) -> {

            try {

                select(conn);

            } catch (SQLException ex) {

                Logger.getLogger(StudentRec.class.getName()).log(Level.SEVERE, null, ex);

            }

        });

        frame.setLayout(new GridLayout(4,1));

        frame.add(tf1);

        frame.add(tf2);

        frame.add(tf3);

        frame.add(display);

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

        frame.setVisible(true);

    }

    private void select(Connection conn) throws SQLException {

        String sql = "select \* from student where rno = 1";

        Statement stmt = conn.createStatement();

        stmt.executeQuery(sql);

        ResultSet rs = stmt.getResultSet();

        while(rs.next()) {

            tf1.setText("       " + rs.getInt("rno"));

            tf2.setText("      " + rs.getString("sname"));

            tf3.setText("      "  + rs.getFloat("per") + "");

        }

    }

}

public class slip10\_2

{

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

        new StudentRec();

    }

}

/\*

 slip no 11 q2 Write a Java program to display information about all columns in the DONAR table

using ResultSetMetaData.

 \*/

package com.mycompany.javaslip;

import java.sql.\*;

public class slip11\_2

{

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

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

        String sql = "select \* from donar";

        Statement stmt = conn.createStatement();

        stmt.executeQuery(sql);

        ResultSet rs = stmt.getResultSet();

        ResultSetMetaData rsmd = rs.getMetaData();

        int colCnt = rsmd.getColumnCount();

        System.out.println("Donar table Meta Data:");

        for(int i=1; i<colCnt; i++) {

            String colName = rsmd.getColumnName(i);

            String colType = rsmd.getColumnTypeName(i);

            int colSize = rsmd.getColumnDisplaySize(i);

            System.out.println(colName + " " + colType + "(" + colSize + ")");

        }

    }

}

/\* slip no 12 \*/

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <h1>Is Perfect?</h1>

        <form action="slip12\_1.jsp" method="post">

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

            <input type="submit" value="is perfect?">

        </form>

        <%

            String numStr = request.getParameter("num");

            int n = 0;

            if(numStr != null && !numStr.isEmpty()) {

                n = Integer.parseInt(numStr);

                if(n > 1) {

                    int sum = 0;

                    for(int i=1; i<=n/2; i++) {

                        if(n % i == 0) {

                            sum += i;

                        }

                    }

                    if(sum == n) {

        %>

                        <h3>Perfect number</h3>

        <%

                    } else {

        %>

                        <h3>Not a perfect number</h3>

        <%

                    }

                }

            }

        %>

    </body>

</html>

/\*

 slip no 12 Q2 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).

 \*/

package com.mycompany.javaslip;

import java.awt.BorderLayout;

import java.sql.\*;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTable;

class ProjectTable {

    private JFrame frame;

    private JTable table;

    ProjectTable() throws SQLException {

        frame = new JFrame("Project Table");

        frame.setLayout(new BorderLayout());

        frame.setSize(600, 150);

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

        createTable(conn);

        insert(conn);

        String[] colNames = {"pid", "pname", "description", "status"};

        String[][] data = retriveData(conn);

        table = new JTable(data, colNames);

        JScrollPane scrPane = new JScrollPane(table);

        frame.getContentPane().add(scrPane, BorderLayout.CENTER);

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

        frame.setVisible(true);

    }

    private void createTable(Connection conn) throws SQLException {

        String sql = "create table if not exists project("

                    + "pid int primary key,"

                    + "pname varchar(30),"

                    + "description varchar(30),"

                    + "status varchar(30))";

        Statement stmt = conn.createStatement();

        stmt.execute(sql);

    }

    private void insert(Connection conn) throws SQLException {

        String sql = "insert into project values"

                    + "(1, 'Game', 'Java Platformer Game', 'complete'),"

                    + "(2, 'Website', 'MERN stack', 'complete'),"

                    + "(3, 'Portfolio', 'PHP', 'complete')";

        Statement stmt = conn.createStatement();

        stmt.executeUpdate(sql);

    }

    private String[][] retriveData(Connection conn) throws SQLException {

        String sql = "select \* from project";

        Statement stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

        ResultSet rs = stmt.executeQuery(sql);

        ResultSetMetaData rsmd = rs.getMetaData();

        int noCol = rsmd.getColumnCount();

        rs.last();

        int noRow = rs.getRow();

        rs.beforeFirst();

        String[][] data = new String[noRow][noCol];

        int rowCnt = 0;

        while (rs.next()) {

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

                data[rowCnt][i - 1] = rs.getString(i);

            rowCnt++;

        }

        return data;

    }

}

public class slip12\_2

{

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

        new ProjectTable();

    }

}

/\*

Slip  no 13 Q1 Write a Java program to display information about the database and list all the tables in

the database. (Use DatabaseMetaData).

 \*/

package com.mycompany.javaslip;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

public class slip13\_1

{

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

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

        DatabaseMetaData md = conn.getMetaData();

        System.out.println("" + md.getDatabaseProductName());

        System.out.println("" + md.getDatabaseProductVersion());

        System.out.println("" + md.getDriverName());

        System.out.println("" + md.getDriverVersion());

        ResultSet tables = md.getTables(null, null, "%", new String[]{"TABLE"});

        System.out.println("Tables in Database:");

        while(tables.next()) {

            String tableName = tables.getString("TABLE\_NAME");

            System.out.println(tableName);

        }

    }

}

/\*

 Slip no13 Q2 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.

 \*/

package com.mycompany.javaslip;

import java.util.Random;

import java.util.logging.Level;

import java.util.logging.Logger;

class ThreadLifeCycle extends Thread {

    private String threadName;

    ThreadLifeCycle(String threadName) {

        this.threadName = threadName;

    }

    public void run() {

        Random rand = new Random();

        int sTime = rand.nextInt(5000);

        System.out.println(threadName + " is created.");

        System.out.println("Sleep time of " + threadName + " is: " + sTime + "ms.");

        try {

            Thread.sleep(sTime);

        } catch (InterruptedException ex) {

            Logger.getLogger(ThreadLifeCycle.class.getName()).log(Level.SEVERE, null, ex);

        }

        System.out.println(threadName + " is dead.");

    }

}

public class slip13\_2

{

    public static void main(String[] args) {

        ThreadLifeCycle t1 = new ThreadLifeCycle("First");

        ThreadLifeCycle t2 = new ThreadLifeCycle("Second");

        ThreadLifeCycle t3 = new ThreadLifeCycle("Third");

        t1.start();

        t2.start();

        t3.start();

    }

}

/\*

slip no 14 Q1 Write a Java program using Multithreading 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.

 \*/

package com.mycompany.javaslip;

import java.io.\*;

import java.util.Scanner;

class SearchThread extends Thread {

    private File file;

    private String searchStr;

    SearchThread(File file, String searchStr) {

        this.file = file;

        this.searchStr = searchStr;

    }

    public void run() {

        searchInFile(file, searchStr);

    }

    public void searchInFile(File file, String searchStr) {

        boolean found = false;

        try (BufferedReader br = new BufferedReader(new FileReader(file))) {

            String line;

            int lineNo = 0;

            while ((line = br.readLine()) != null) {

                lineNo++;

                if (line.contains(searchStr)) {

                    System.out.println("Found '" + searchStr + "' in " + file.getName() + " at line " + lineNo);

                    found = true;

                }

            }

        } catch (IOException ex) {

            System.err.println("Error reading file: " + file.getName());

        }

        if (!found) {

            System.out.println(searchStr + " not found in " + file.getName());

        }

    }

}

public class slip14\_1

{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter string to be searched in files:");

        String searchStr = sc.nextLine();

        File currDir = new File(".");

        File[] files = currDir.listFiles();

        if (files != null) {

            boolean foundInAnyFile = false;

            for (File file : files) {

                if (file.isFile() && file.getName().endsWith(".txt")) {

                    SearchThread t = new SearchThread(file, searchStr);

                    t.start();

                    foundInAnyFile = true;

                }

            }

            if (!foundInAnyFile) {

                System.out.println("No text files found in the current directory.");

            }

        } else {

            System.err.println("Error: Unable to access current directory.");

        }

    }

}

/\* slipno 14 Q2 \*/

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

        <style>

            .res { color: red; font-size: 18px; }

        </style>

    </head>

    <body>

        <h1>Calculate sum of fist and last digit?</h1>

        <form action="slip14\_2.jsp" method="post">

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

            <input type="submit" value="sum?">

        </form>

        <%

            String numStr = request.getParameter("num");

            int n = 0;

            if(numStr != null && !numStr.isEmpty()) {

                n = Integer.parseInt(numStr);

                int fDigit = n;

                while(fDigit >= 10) {

                    fDigit /= 10;

                }

                int lDigit = n % 10;

                int sum = fDigit + lDigit;

        %>

                <h3 class="res">Sum of first and last digit is <%= sum %></h3>

        <%

            }

        %>

    </body>

</html>

/\*

 slip no 15 q1 Write a java program to display name and priority of a Thread.

 \*/

package com.mycompany.javaslip;

class MyThread extends Thread {

    public void run() {

        System.out.println("Name of the thread: " + Thread.currentThread().getName());

        System.out.println("Priority of the thread: " + Thread.currentThread().getPriority());

    }

}

public class slip15\_1

{

    public static void main(String[] args) {

        MyThread t1 = new MyThread();

        MyThread t2 = new MyThread();

        t1.start();

        t2.start();

    }

}

/\*

slip no 16 Q1. Write a java program to create a TreeSet, add some colors (String) and print out the

content of TreeSet in ascending order

 \*/

package com.mycompany.javaslip;

import java.util.\*;

public class slip16\_1

{

    public static void main(String[] args) {

        Set<String> colors = new TreeSet<>();

        colors.add("Red");

        colors.add("Blue");

        colors.add("Green");

        colors.add("Yellow");

        colors.add("Black");

        System.out.println(colors);

    }

}

/\*

 slip no 16 Q2 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)

 \*/

package com.mycompany.javaslip;

import java.sql.\*;

import java.util.Scanner;

class Teacher {

    Teacher() throws SQLException, SQLException {

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

        for(int i=0; i<5; i++)

            insert(conn);

        select(conn);

    }

    private void insert(Connection conn) throws SQLException {

        String sql = "insert into teacher values(?, ?, ?)";

        PreparedStatement ps = conn.prepareStatement(sql);

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter tno:");

        ps.setInt(1, sc.nextInt());

        sc.nextLine();

        System.out.println("Enter tname:");

        ps.setString(2, sc.nextLine());

        System.out.println("Enter subject:");

        ps.setString(3, sc.nextLine());

        ps.executeUpdate();

    }

    private void select(Connection conn) throws SQLException {

        String sql = "select \* from teacher where subject = 'java'";

        Statement stmt = conn.createStatement();

        ResultSet rs = stmt.executeQuery(sql);

        while(rs.next()) {

            System.out.println("teacher tno: " + rs.getInt("tno"));

            System.out.println("teacher tname: " + rs.getString("tname"));

            System.out.println("teacher subject: " + rs.getString("subject"));

        }

    }

}

public class slip16\_2

{

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

        new Teacher();

    }

}

/\*

Slip no 17  q1Write 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.

 \*/

package com.mycompany.javaslip;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class slip17\_1

{

    public static void main(String[] args) {

        Set<Integer> set = new TreeSet<>();

        Scanner sc = new Scanner(System.in);

        System.out.println("How many integers:");

        int n = sc.nextInt();

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

        for(int i=0; i<n; i++)

            set.add(sc.nextInt());

        System.out.println(set);

    }

}

/\*

Slip no 17 Q2 Write a java program using Multithreading to display the number’s between 1 to 100

continuously in a JTextField by clicking on JButton. (Use Runnable Interface &

Swing).

 \*/

package com.mycompany.javaslip;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.util.logging.\*;

import javax.swing.\*;

public class slip17\_2

{

    private JFrame frame;

    private JTextField tf;

    private JButton print;

    private Thread intThread;

    slip17\_2() {

        frame = new JFrame("Integer printing App");

        frame.setSize(300, 200);

        frame.setLayout(new GridLayout(2,1));

        tf = new JTextField();

        print = new JButton("Print");

        frame.add(tf);

        frame.add(print);

        print.addActionListener((ActionEvent e) -> {

            tf.setText("");

            if(intThread == null || !intThread.isAlive()) {

                intThread = new Thread(new Runnable() {

                    @Override

                    public void run() {

                        while(true) {

                            for(int i=1; i<=100; i++) {

                                tf.setText(String.valueOf(i));

                                try {

                                    Thread.sleep(500);

                                } catch (InterruptedException ex) {

                                    Logger.getLogger(S17Q2.class.getName()).log(Level.SEVERE, null, ex);

                                }

                            }

                            tf.setText("");

                        }

                    }

                });

                intThread.start();

            }

        });

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

        frame.setVisible(true);

    }

    public static void main(String[] args) {

        new S17Q2();

    }

}

/\*

 Slip n 18 q1 Write a java program using Multithreading to display all the vowels from a given

String. Each vowel should be displayed after every 3 seconds.

 \*/

package com.mycompany.javaslip;

import java.util.Scanner;

import java.util.logging.\*;

public class slip18\_1

{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter any string:");

        String str = sc.nextLine();

        Thread t = new Thread(() -> {

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

                String str2 = str.toLowerCase();

                char ch = str2.charAt(i);

                if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

                    System.out.println(ch);

                    try {

                        Thread.sleep(3000);

                    } catch (InterruptedException ex) {

                        Logger.getLogger(slip18\_1.class.getName()).log(Level.SEVERE, null, ex);

                    }

                    System.out.println("3 seconds are passed....");

                }

            }

        });

        t.start();

    }

}

/\*

 slip no 19 Q1 Write a java program to accept ‘N’ Integers from a user store them into LinkedList

Collection and display only negative integers.

 \*/

package com.mycompany.javaslip;

import java.util.\*;

public class slip19\_1

{

    public static void main(String[] args) {

        List<Integer> l = new LinkedList<>();

        Scanner sc = new Scanner(System.in);

        System.out.println("How many values:");

        int n = sc.nextInt();

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

        for(int i=0; i<n; i++)

            l.add(sc.nextInt());

        System.out.println("Negative integers are:");

        Iterator itr = l.iterator();

        while(itr.hasNext()) {

            int num = (int)itr.next();

            if(num < 0)

                System.out.println(num);

        }

    }

}

/\* Slip no 20\*/

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <form action="slip20\_1.jsp" method="post">

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

            <input type="submit" value="show in words">

        </form>

        <%

        String numStr = request.getParameter("num");

        if(numStr != null && !numStr.isEmpty()) {

            int t = Integer.parseInt(numStr);

            int rev = 0, rem;

            // reverse the number

            while(t > 0) {

                rem = t % 10;

                rev = (rev \* 10) + rem;

                t = t / 10;

            }

            t = rev;

            rev = 0;

            while(t > 0) {

                rem = t % 10;

                rev = (rev \* 10) + rem;

                t = t / 10;

                switch(rem) {

                    case 0: out.println("zero");

                        break;

                    case 1: out.println("one");

                        break;

                    case 2: out.println("two");

                        break;

                    case 3: out.println("three");

                        break;

                    case 4: out.println("four");

                        break;

                    case 5: out.println("five");

                        break;

                    case 6: out.println("six");

                        break;

                    case 7: out.println("seven");

                        break;

                    case 8: out.println("eight");

                        break;

                    case 9: out.println("nine");

                        break;

                }

            }

        }

        %>

    </body>

</html>

/\*

 slip no 20 q2Write a java program using Multithreading to demonstrate drawing temple (Use

Swing)

 \*/

package com.mycompany.javaslip;

import javax.swing.\*;

import java.awt.\*;

class TempleDrawing extends JFrame

{

    public TempleDrawing()

 {

        setTitle("Simple Temple Drawing");

        setSize(300, 300);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLocationRelativeTo(null);

        TemplePanel templePanel = new TemplePanel();

        add(templePanel);

        setVisible(true);

    }

}

class TemplePanel extends JPanel

 {

    @Override

    protected void paintComponent(Graphics g)

 {

        super.paintComponent(g);

        drawTemple(g);

    }

    private void drawTemple(Graphics g)

  {

        g.setColor(Color.BLACK);

        g.fillRect(100, 100, 100, 100); // Main structure

        g.setColor(Color.WHITE);

        g.fillRect(130, 150, 40, 50); // Main Door

        g.setColor(Color.RED);

        int[] xPoints = {100, 150, 200}; // Triangle for roof

        int[] yPoints = {100, 50, 100};

        g.fillPolygon(xPoints, yPoints, 3);

        g.setColor(Color.ORANGE);

        g.fillRect(150, 40, 20, 10); // Flag

    }

}

public class slip20\_2

{

    public static void main(String[] args)

 {

        SwingUtilities.invokeLater(() ->

        {

            new TempleDrawing();

        });

    }

}

/\*

slip no 21 Q1. Write a java program to accept ‘N’ Subject Names from a user store them into

LinkedList Collection and Display them by using Iterator interface.

 \*/

package com.mycompany.javaslip;

import java.util.\*;

public class slip21\_1

{

    public static void main(String[] args) {

        List<String> l = new LinkedList<>();

        Scanner sc = new Scanner(System.in);

        System.out.println("How many subjects:");

        int n = sc.nextInt();

        sc.nextLine();

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

        for(int i=0; i<n; i++)

            l.add(sc.nextLine());

        System.out.println("Subjects are:");

        Iterator itr = l.iterator();

        while(itr.hasNext()) {

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

        }

    }

}

/\*

slip no 22 Q2 Write a java program using Multithreading 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)

 \*/

package com.mycompany.javaslip;

import java.util.LinkedList;

class SharedResource {

    private LinkedList<String> buffer = new LinkedList<>();

    private int capacity = 1;

    public synchronized void produce(String value) {

        while(buffer.size() == capacity) {

            try {

                wait();

            } catch(InterruptedException e) {

                e.printStackTrace();

            }

        }

        buffer.add(value);

        System.out.println("Produced: " + value);

        notifyAll();

    }

    public synchronized String consume() {

        while(buffer.size() == 0) {

            try {

                wait();

            } catch(InterruptedException e) {

                e.printStackTrace();

            }

        }

        String value = buffer.removeFirst();

        System.out.println("Consume: " + value);

        notifyAll();

        return value;

    }

}

class Producer extends Thread {

    private SharedResource sharedResource;

    public Producer(SharedResource sharedResource) {

        this.sharedResource = sharedResource;

    }

    @Override

    public void run() {

        for(int i=0; i<5; i++) {

            String value = "Value " + i;

            sharedResource.produce(value);

            try {

                sleep(1000);

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

    }

}

class Consumer extends Thread {

    private SharedResource sharedResource;

    public Consumer(SharedResource sharedResource) {

        this.sharedResource = sharedResource;

    }

    @Override

    public void run() {

        for(int i=0; i<5; i++) {

            String value = "Value " + i;

            sharedResource.consume();

            try {

                sleep(1000);

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

    }

}

public class slip21\_2

{

    public static void main(String[] args) {

        SharedResource sharedResource = new SharedResource();

        Producer producer = new Producer(sharedResource);

        Consumer consumer = new Consumer(sharedResource);

        producer.start();

        consumer.start();

    }

}

/\*

slip no 22 Q1 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

 \*/

package com.mycompany.javaslip;

import java.sql.\*;

import java.util.Scanner;

public class slip22\_1

{

    private static void insert(Connection conn) throws SQLException {

        String sql = "insert into emp2 values (?, ?, ?)";

        PreparedStatement ps = conn.prepareStatement(sql);

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter eno:");

        ps.setInt(1, sc.nextInt());

        sc.nextLine();

        System.out.println("Enter ename:");

        ps.setString(2, sc.nextLine());

        System.out.println("Enter salary:");

        ps.setFloat(3, sc.nextFloat());

        ps.executeUpdate();

    }

    private static void update(Connection conn) throws SQLException {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter eno:");

        int eno = sc.nextInt();

        sc.nextLine();

        System.out.println("Enter new  ename:");

        String ename = sc.nextLine();

        System.out.println("Enter new salary:");

        float salary = sc.nextFloat();

        String sql = "update emp2 set ename = '" + ename + "', salary = " + salary + " where eno = " + eno;

        Statement stmt = conn.createStatement();

        stmt.executeUpdate(sql);

    }

    private static void display(Connection conn) throws SQLException {

        String sql = "select \* from emp2";

        Statement stmt = conn.createStatement();

        ResultSet rs = stmt.executeQuery(sql);

        System.out.println("Emp table data:");

        while (rs.next()) {

            System.out.println("eno: " + rs.getInt("eno"));

            System.out.println("ename: " + rs.getString("ename"));

            System.out.println("salary: " + rs.getFloat("salary"));

        }

    }

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

        Scanner sc = new Scanner(System.in);

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

        int ch;

        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.println("-------------------------");

            System.out.println("Enter your choice:");

            ch = sc.nextInt();

            switch (ch) {

                case 1:

                    insert(conn);

                    break;

                case 2:

                    update(conn);

                    break;

                case 3:

                    display(conn);

                        break;

            }

        } while (ch != 4);

    }

}

\*/ slip no 22 Q2 \*/

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@page import="java.time.LocalTime" %>

<!DOCTYPE html>

<html>

    <head>

        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

        <title>JSP Page</title>

    </head>

    <body>

        <form action="slip22\_2.jsp" method="post">

            Enter user name :<input type="text" name="user"><br>

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

        </form>

        <%

            String user = request.getParameter("user");

            if(user != null && !user.isEmpty()) {

                LocalTime currTime = LocalTime.now();

                int hour = currTime.getHour();

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

                    out.println("Good Morning " + user);

                else if(hour >= 12 && hour <= 18)

                    out.println("Good Afternoon " + user);

                else

                    out.println("Good Morning " + user);

            }

        %>

    </body>

</html>

/\*

slip no 23 Q1 Write a java program using Multithreading to accept a String from a user and display

each vowel from a String after every 3 seconds

 \*/

package com.mycompany.javaslip;

import java.util.Scanner;

import java.util.logging.\*;

public class slip23\_1

{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter any string:");

        String str = sc.nextLine();

        Thread t = new Thread(() -> {

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

                String str2 = str.toLowerCase();

                char ch = str2.charAt(i);

                if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

                    System.out.println(ch);

                    try {

                        Thread.sleep(3000);

                    } catch (InterruptedException ex) {

                        Logger.getLogger(slip23\_1.class.getName()).log(Level.SEVERE, null, ex);

                    }

                    System.out.println("3 seconds are passed....");

                }

            }

        });

        t.start();

    }

}

/\*

 Slip no 24 Q1 Write a java program using Multithreading to scroll the text from left to right

continuously (Use Swing).

 \*/

package com.mycompany.javaslip;

import javax.swing.\*;

class TextScrolling extends JFrame implements Runnable {

    private JLabel label;

    private String text;

    private Thread thread;

    public TextScrolling(String text) {

        this.text = text;

        label = new JLabel(text);

        add(label);

        setSize(300, 100);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setVisible(true);

    }

    public void startScrolling() {

        thread = new Thread(this);

        thread.start();

    }

    @Override

    public void run() {

        try {

            while (true) {

                String labelText = label.getText();

                labelText = labelText.substring(1) + labelText.charAt(0);

                label.setText(labelText);

                Thread.sleep(200); // Adjust scrolling speed

            }

        } catch (InterruptedException e) {

            e.printStackTrace();

        }

    }

}

public class slip24\_1

{

    public static void main(String[] args) {

        SwingUtilities.invokeLater(() -> {

            TextScrolling ts = new TextScrolling("Hello, this text is scrolling continuously!");

            ts.startScrolling();

        });

    }

}

/\*

SLip no 25 Q2 Write a Java Program for the following: Assume database is already created.

 \*/

package com.mycompany.javaslip;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class slip25\_2

{

    JFrame frame;

    JButton b1, b2, b3;

    JTextField tf;

    slip25\_2() throws SQLException {

        frame = new JFrame("DB App");

        frame.setLayout(new BorderLayout());

        frame.setSize(600, 100);

        JPanel p1 = new JPanel();

        JPanel p2 = new JPanel();

        tf = new JTextField();

        p1.setLayout(new GridLayout(1, 2));

        p1.add(new JLabel("Type your DDL query:"));

        p1.add(tf);

        b1 = new JButton("Create Table");

        b2 = new JButton("Alter Table");

        b3 = new JButton("Drop Table");

        p2.setLayout(new GridLayout(1, 3));

        p2.add(b1);

        p2.add(b2);

        p2.add(b3);

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

        b1.addActionListener((ActionEvent e) -> {

            try {

                create(conn);

            } catch (SQLException ex) {

                Logger.getLogger(S25Q2.class.getName()).log(Level.SEVERE, null, ex);

            }

        });

        b2.addActionListener((ActionEvent e) -> {

            try {

                alter(conn);

            } catch (SQLException ex) {

                Logger.getLogger(S25Q2.class.getName()).log(Level.SEVERE, null, ex);

            }

        });

        b3.addActionListener((ActionEvent e) -> {

            try {

                drop(conn);

            } catch (SQLException ex) {

                Logger.getLogger(S25Q2.class.getName()).log(Level.SEVERE, null, ex);

            }

        });

        frame.add(p1, BorderLayout.CENTER);

        frame.add(p2, BorderLayout.SOUTH);

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

        frame.setVisible(true);

    }

    private void create(Connection conn) throws SQLException {

        String sql = tf.getText();

        Statement stmt = conn.createStatement();

        stmt.execute(sql);

    }

    private void alter(Connection conn) throws SQLException {

        String sql = tf.getText();

        Statement stmt = conn.createStatement();

        stmt.execute(sql);

    }

    private void drop(Connection conn) throws SQLException {

        String sql = tf.getText();

        Statement stmt = conn.createStatement();

        stmt.execute(sql);

    }

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

        new S25Q2();

    }

}

/\*

Slip no 26 Q1 Write a Java program to delete the details of given employee (ENo EName Salary).

Accept employee ID through command line. (Use PreparedStatement Interface)

 \*/

package com.mycompany.javaslip;

import java.sql.\*;

public class slip26\_1

{

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

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

        String sql = "delete from emp where id = ?";

        PreparedStatement ps = conn.prepareStatement(sql);

        ps.setInt(1, Integer.parseInt(args[0]));

        ps.executeUpdate();

    }

}

/\*

slip no 27 Q1 Write a Java Program to display the details of College (CID, CName, address, Year)

database table on JTable.

 \*/

package com.mycompany.javaslip;

import java.awt.BorderLayout;

import java.sql.\*;

import javax.swing.\*;

class CollegeTable {

    private JFrame frame;

    private JTable table;

    CollegeTable() throws SQLException {

        frame = new JFrame("Project Table");

        frame.setLayout(new BorderLayout());

        frame.setSize(600, 150);

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

        String[] colNames = {"cid", "cname", "address", "year"};

        String[][] data = retriveData(conn);

        table = new JTable(data, colNames);

        JScrollPane scrPane = new JScrollPane(table);

        frame.getContentPane().add(scrPane, BorderLayout.CENTER);

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

        frame.setVisible(true);

    }

    private String[][] retriveData(Connection conn) throws SQLException {

        String sql = "select \* from college";

        Statement stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

        ResultSet rs = stmt.executeQuery(sql);

        ResultSetMetaData rsmd = rs.getMetaData();

        int noCol = rsmd.getColumnCount();

        rs.last();

        int noRow = rs.getRow();

        rs.beforeFirst();

        String[][] data = new String[noRow][noCol];

        int rowCnt = 0;

        while (rs.next()) {

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

                data[rowCnt][i - 1] = rs.getString(i);

            rowCnt++;

        }

        return data;

    }

}

public class slip27\_1

{

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

        new CollegeTable();

    }

}

/\*

Slip no 28 Q2 Write a java program to display name of currently executing Thread in multithreading

 \*/

package com.mycompany.javaslip;

public class slip28\_2

{

    public static void main(String[] args) {

        Thread t = new Thread(() -> {

            System.out.println("Name of the thread: " + Thread.currentThread().getName());

        });

        t.start();

    }

}

/\*

Slip no 29 Q1. Write a Java program to display information about all columns in the DONAR table

using ResultSetMetaData.

 \*/

package com.mycompany.javaslip;

import java.sql.\*;

public class slip29\_1

{

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

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

        String sql = "select \* from donar";

        Statement stmt = conn.createStatement();

        stmt.executeQuery(sql);

        ResultSet rs = stmt.getResultSet();

        ResultSetMetaData rsmd = rs.getMetaData();

        int colCnt = rsmd.getColumnCount();

        System.out.println("Donar table Meta Data:");

        for(int i=1; i<colCnt; i++) {

            String colName = rsmd.getColumnName(i);

            String colType = rsmd.getColumnTypeName(i);

            int colSize = rsmd.getColumnDisplaySize(i);

            System.out.println(colName + " " + colType + "(" + colSize + ")");

        }

    }

}

/\*

slip no 29 Q2. 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

 \*/

package com.mycompany.javaslip;

import java.util.\*;

public class slip29\_2

{

    public static void main(String[] args) {

        List<Integer> l = new LinkedList<>();

        Scanner sc = new Scanner(System.in);

        int ch;

        do {

            System.out.println("Menu");

            System.out.println("1. Insert at head");

            System.out.println("2. Delete tail.");

            System.out.println("3. Display size");

            System.out.println("4. Exit");

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

            System.out.println("Enter your choice:");

            ch = sc.nextInt();

            System.out.println();

            switch(ch) {

                case 1: System.out.println("Enter a number:");

                    l.addFirst(sc.nextInt());

                    break;

                case 2: l.removeLast();

                    break;

                case 3:

                    System.out.println("Size : " + l.size() + "\n" + l);

                    break;

                default: System.out.println("Invalid choice.");

            }

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

        } while(ch != 4);

    }

}

/\*

Slip no 30 Q1. Write a java program using Multithreading to demonstrate drawing Indian flag (Use

Swing

 \*/

package com.mycompany.javaslip;

import javax.swing.\*;

import java.awt.\*;

class IndianFlag extends JFrame {

    public IndianFlag() {

        setTitle("Simple Temple Drawing");

        setSize(300, 300);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLocationRelativeTo(null);

        FlagPanel flagPanel = new FlagPanel();

        add(flagPanel);

        setVisible(true);

    }

}

class FlagPanel extends JPanel {

    @Override

    protected void paintComponent(Graphics g) {

        super.paintComponent(g);

        drawFlag(g);

    }

    private void drawFlag(Graphics g) {

        g.setColor(Color.ORANGE);

        g.fillRect(50, 50, 200, 50);

        g.setColor(Color.WHITE);

        g.fillRect(50, 100, 200, 50);

        g.setColor(Color.GREEN);

        g.fillRect(50, 150, 200, 50);

    }

}

public class slip30\_1

{

    public static void main(String[] args) {

        SwingUtilities.invokeLater(() -> {

            new IndianFlag();

        });

    }

}