Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Department of Multidisciplinary Engineering

Lab Manual

Course Code	Course Name	Teaching Scheme (Hrs. /Week)	Credits
CS2218:	Object Oriented Programming	3 Theory, 2 Lab,1 Tut	5

Class: -SY	Branch: IT	
Year: -2022-23	Prepared By:	Prof. (Dr)Ajay Talele, Prof. D G Kanade
S/W Requirement: JDK, Eclipse		

Vision and Mission of Institute

VISION

"To be Globally Acclaimed Institute in Technical Education and Research for Holistic Socio-Economic Development"

MISSION

- To ensure that 100% students are employable and employed in Industry, Higher Studies, Become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.
- To Strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment
  And Faculty Competence
- Promote Research Culture among Students and Faculty through Projects and Consultancy.
- To make students Socially Responsible Citizen

Course Outcomes

CO_1: Understand object-oriented programming features

CO_2: Develop real world applications using class, inheritance and polymorphism.

CO_3: Adapt Best Practices of Class Design by using Standard Templates Library

CO_4: Solve computing problems by applying the knowledge of Exception handling and Multithreading.

CO_5: Design solutions by choosing suitable data structures such as Array, Vector, Map etc

CO_6: Implement applications using Java I/O and event-based GUI handling principles

INDEX

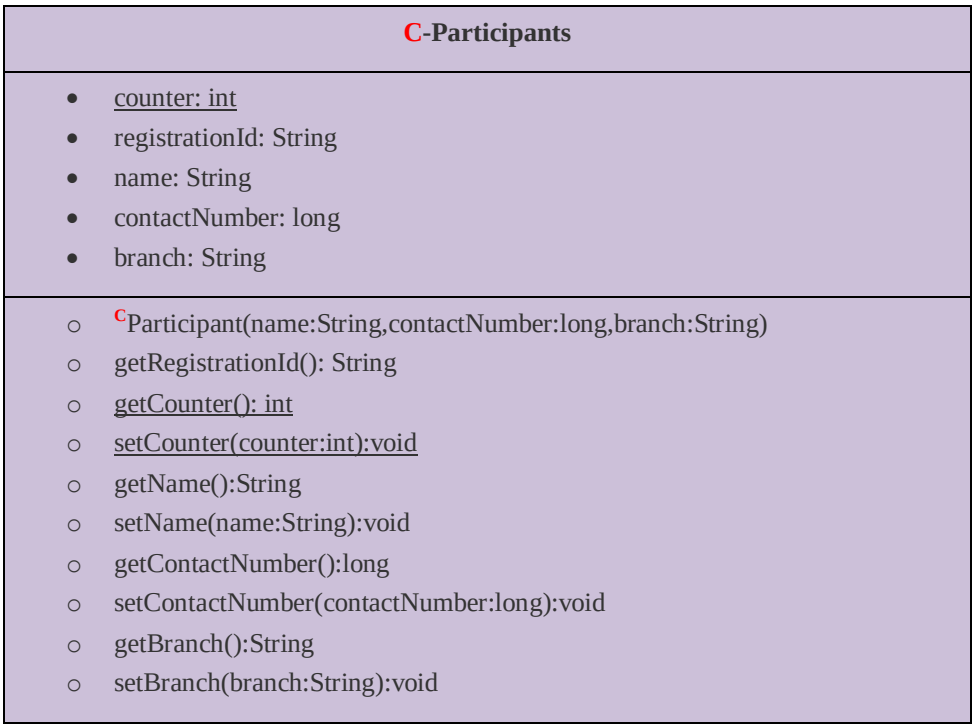
Assignment No.1

Name of Student: Kartik Banshi Katkar

Batch: 2 **Branch: Information Technology** **Roll No: 29**

Problem Statement

A VIT Melange committee is conducting auditions to admit interested candidates. You need to implement a Participant class for the dance club based on the class diagram and description given below.



Method Description

Participant (String name, long contactNumber, String branch)

- Initialize the name, contactNumber and branch instance variables appropriately with the values passed to the constructor.
- Generate the registrationId using the static variable counter.
- The value of registrationId should start from 'D1001' and the numerical part should be incremented by 1 for the subsequent values.
- Initialize the counter in static block.

Implement the appropriate getter and setter methods.

Test the functionalities using the provided Tester class. Create two or more Participant objects and validate that the values of the member variables are proper.

Sample Input and Output

For constructor

Input

For first Participant object

Parameters	**Values**
name	Rohit
contactNumber	1234567889
branch	Computer

For second Participant object

Parameters	**Values**
name	Sayli
contactNumber	1988612300
branch	Mechanical

Expected Output

Hi Rohit! Your registration id is D1001

Hi Sayli! Your registration id is D1002

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class Participant {
    private static int count = 1000;
    private String name;
    private long contactNumber;
    private String branch;
    private int registrationId;


    // Add constructor with name, contactNumber, and branch arguments
```

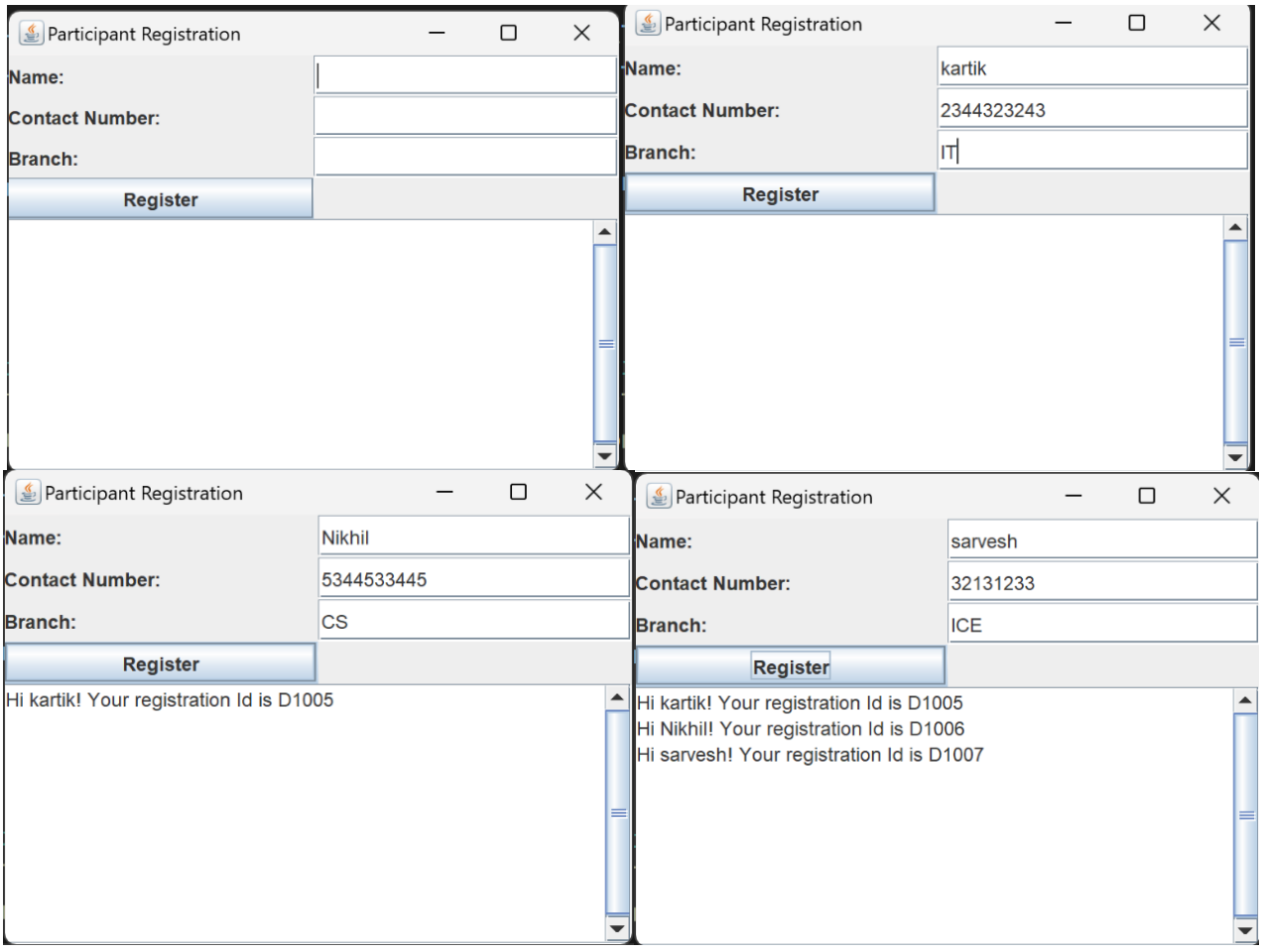
```
public Participant(String name, long contactNumber, String branch) {
        this.name = name;
        this.contactNumber = contactNumber;
        this.branch = branch;
        count++;
        registrationId = count;
    }

    // Add copy constructor that takes a Participant object as argument
    public Participant(Participant p) {
        this(p.name, p.contactNumber, p.branch);
    }

    public String getName() {
        return name;
    }

    public long getContactNumber() {
        return contactNumber;
    }

    public String getBranch() {
        return branch;
    }

    public int getregistrationId() {
        return registrationId;
    }
}

// Rename class to ParticipantGUI with capital 'P'
```

```
class ParticipantGUI extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JPanel panel;
    private JLabel nameLabel, contactLabel, branchLabel;
    private JTextField nameField, contactField, branchField;
    private JButton addButton;
    private JTextArea outputArea;
    private JScrollPane scrollPane;
    private Participant p1, p2, p3, p4;

    public ParticipantGUI() {
        super("Participant Registration");

        // create participant objects using the new constructor
        p1 = new Participant("Rohit", 123456789, "Computer");
        p2 = new Participant("Sayali", 1988612300, "Mechanical");
        p3 = new Participant(p2);
        p4 = new Participant("Neel", 91376, "ICE");

        // create GUI components
        panel = new JPanel(new GridLayout(4, 2));
        nameLabel = new JLabel("Name:");
        contactLabel = new JLabel("Contact Number:");
        branchLabel = new JLabel("Branch:");
        nameField = new JTextField(20);
        contactField = new JTextField(10);
        branchField = new JTextField(20);
        addButton = new JButton("Register");
        addButton.addActionListener(this);
        outputArea = new JTextArea(10, 30);
        outputArea.setEditable(false);
```

```
        scrollPane = new JScrollPane(outputArea);

        // add components to panel
        panel.add(nameLabel);
        panel.add(nameField);
        panel.add(contactLabel);
        panel.add(contactField);
        panel.add(branchLabel);
        panel.add(branchField);
        panel.add(addButton);

        // add components to frame
        add(panel, BorderLayout.NORTH);
        add(scrollPane, BorderLayout.CENTER);

        // set frame properties
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        long contact = Long.parseLong(contactField.getText());
        String branch = branchField.getText();
        Participant p = new Participant(name, contact, branch);
        outputArea.append("Hi " + p.getName() + "! Your registration Id is D" + p.getregistrationId() + "\n");
    }
}
public class gui1 {
    public static void main(String[] args) {
```

```
        new ParticipantGUI();
    }
}
```

Screenshots/Results:



Assignment No. 2

Name of Student: Kartik Banshi Katkar

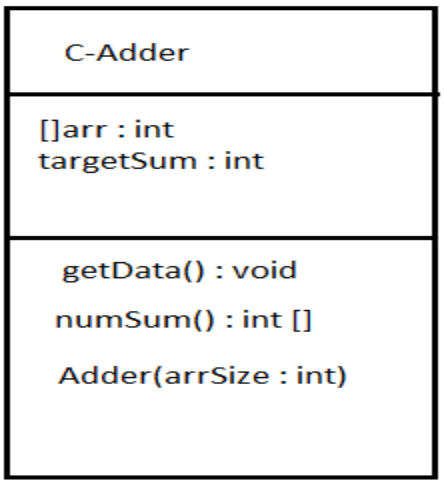
Batch: 2 **Branch: IT** **Roll No: 29**

Problem Statement

There is a class Adder which has two data members of type 1D int array and int variable. It has two functions: getdata and numsum. Function getdata accepts non-empty array of distinct integers from user in 1D int array data member and a targetsum in another data member.

The function numsum adds any two elements from an input array which is equal to targetsum and return an array of resulting two elements, in any order. If no two numbers sum up to the target sum, the function should return an empty array.

Note that the target sum is to be obtained by summing two different integers in the array; you can't add a single integer to itself in order to obtain the target sum. You can assume that there will be at most one pair of numbers summing up to the target sum. Use constructor. Use extra variables if needed.



Sample Input and Output

Test Case 1

Input Parameters	Values	Expected Output
1D Array	[3,5,-4,8,11,1,-1,7]	[8,7]
targetsum	15	

Test Case 2

Input Parameters	Values	Expected Output
1D Array	[3,5, -4,8,11,1, -1,6]	[ ]
targetsum	15	

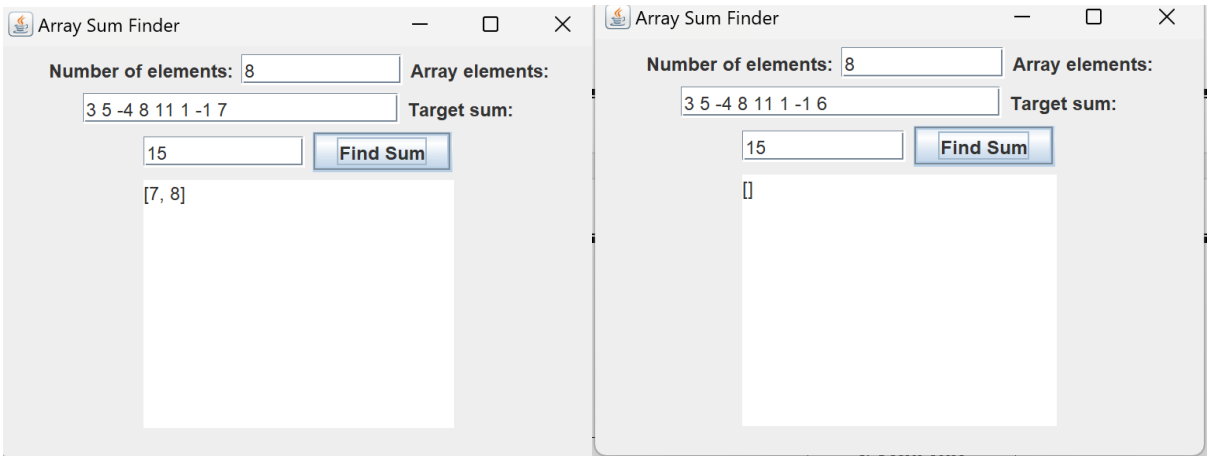
Code:

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


import javax.swing.*;


public class gui2 extends JFrame {


    private JLabel numLabel, elemLabel, sumLabel;
    private JTextField numField, elemField, sumField;
    private JButton button;
    private JTextArea resultArea;


    public gui2() {
        super("Array Sum Finder");


        numLabel = new JLabel("Number of elements:");
        elemLabel = new JLabel("Array elements:");
        sumLabel = new JLabel("Target sum:");
        numField = new JTextField(10);
        elemField = new JTextField(20);
        sumField = new JTextField(10);
        button = new JButton("Find Sum");
        resultArea = new JTextArea(10, 20);
```

```
    JPanel panel = new JPanel();
    panel.add(numLabel);
    panel.add(numField);
    panel.add(elemLabel);
    panel.add(elemField);
    panel.add(sumLabel);
    panel.add(sumField);
    panel.add(button);
    panel.add(resultArea);

    add(panel);

    button.addActionListener(e -> {
        int num = Integer.parseInt(numField.getText());
        int[] arr = new int[num];
        String[] elements = elemField.getText().split(" ");
        for (int i = 0; i < num; i++) {
            arr[i] = Integer.parseInt(elements[i]);
        }
        int targetsum = Integer.parseInt(sumField.getText());
        ArrayList result = findTargetSum(arr, targetsum);
        resultArea.setText(result.toString());
    });

    setSize(400, 300);
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public ArrayList findTargetSum(int[] arr, int targetsum) {
    ArrayList result=new ArrayList<>();
```

```
        for (int i = 0; i < (arr.length-1); i++) {
            int temp = arr[i];
            int sub = targetsum - temp;
            for (int j = i+1; j < arr.length; j++) {
                if(sub == arr[j])
                {
                    result.add(arr[j]);
                    result.add(arr[i]);
                }
            }
        }
        return result;
    }

    public static void main(String[] args) {
        new gui2();
    }
}
```

Results:



Assignment No. 3

GRN: 12111367 **Name of the Student: Kartik Banshi Katkar** **Roll No.: 29**
Class: SY-IT **Division: A** **Batch: 2**

Problem Statement

Calculate area of triangle, square & circle using function overloading. Function parameter accept from user. Create Base Class **Shape** and Derived Classes **Triangle, Square, Circle** respectively. Implement **getInputs()** Method for accepting inputs, and Overload **setArea()** method for calculating area of respective shapes.

Use Class **Tester** for creating objects.

Sample Input and Output

Sample Input/Parameter for Triangle	Values	Expected Output
Height (H)	50	2500
Base (B)	100	

Sample Input/Parameter for Circle	Values	Expected Output
π (Pie)	3.14	7850
Radius (R)	50	

Sample Input/Parameter for Square	Values	Expected Output
Side (S)	15	225

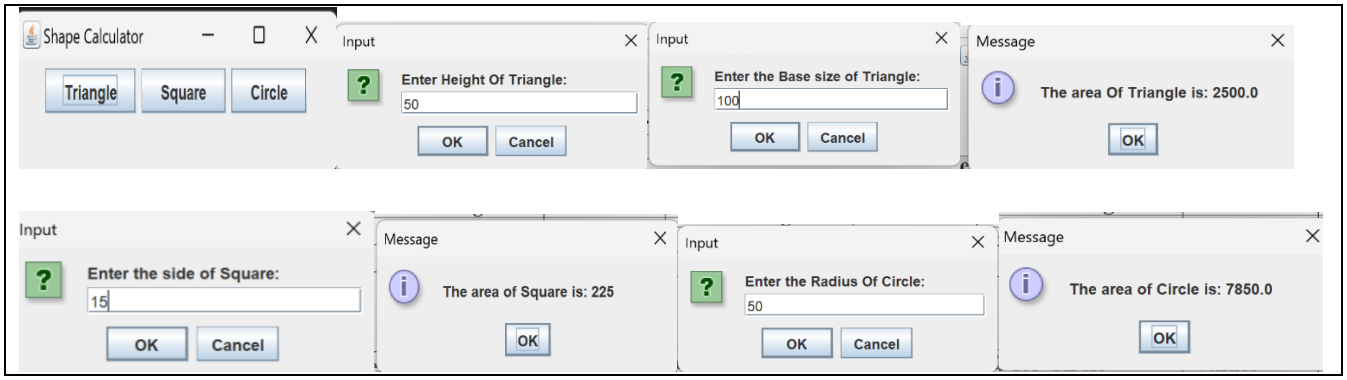
Code:

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

import javax.swing.*;

class Triangle extends Shape {
   float height, base;

   void getInput() {
      String s1 = JOptionPane.showInputDialog("Enter Height Of Triangle:");
      height = Float.parseFloat(s1);

      String s2 = JOptionPane.showInputDialog("Enter the Base size of Triangle:");
```

```
        base = Float.parseFloat(s2);
    }

    void setArea() {
        float area;
        area = 0.5f * height * base;
        JOptionPane.showMessageDialog(null, "The area Of Triangle is: " + area);
    }
}

class Square extends Shape {
    int side;

    void getInput() {
        String s = JOptionPane.showInputDialog("Enter the side of Square:");
        side = Integer.parseInt(s);
    }

    void setArea() {
        long area;
        area = side * side;
        JOptionPane.showMessageDialog(null, "The area of Square is: " + area);
    }
}

class Circle extends Shape {
    double radius;

    void getInput() {
        String s = JOptionPane.showInputDialog("Enter the Radius Of Circle:");
        radius = Double.parseDouble(s);
    }

    void setArea() {
        double area;
        area = 3.14 * radius * radius;
        JOptionPane.showMessageDialog(null, "The area of Circle is: " + area);
    }
}

class Shape {
    void getInput() {
        System.out.println();
    }
}

public class gui3 extends JFrame {
    JButton triangleButton, squareButton, circleButton;
    JPanel panel;

    public gui3() {
        panel = new JPanel();
        triangleButton = new JButton("Triangle");
        squareButton = new JButton("Square");
        circleButton = new JButton("Circle");

        panel.add(triangleButton);
        panel.add(squareButton);
        panel.add(circleButton);

        add(panel);
```

```
        triangleButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Triangle obj1 = new Triangle();
                obj1.getInput();
                obj1.setArea();
            }
        });

        squareButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Square obj2 = new Square();
                obj2.getInput();
                obj2.setArea();
            }
        });

        circleButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Circle obj3 = new Circle();
                obj3.getInput();
                obj3.setArea();
            }
        });

        setTitle("Shape Calculator");
        setSize(300, 100);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public static void main(String[] args) {
        new gui3();
    }
}
```

Screenshots/Results:



Assignment No. 4

Name of Student: Kartik Banshi Katkar

Batch: 2 **Branch: IT** **Roll No: 29**

Problem Statement

Write a Java program for following exception, develop a suitable scenario in which the following exceptions occur:

a. divide by zero
b. Array index out of bounds exception
c. Null pointer Exception

Sample Input and Output

Sample Input/Parameter	Expected Output
Try to divide a number by zero	You shouldn't divide a number by zero.
Try to access the array index which does not exist.	OOPs!!!Array Index 7 out of bounds for length 6.
Try to find the length of String in method (pass parameter string as null)	Null Pointer Exception arises!!

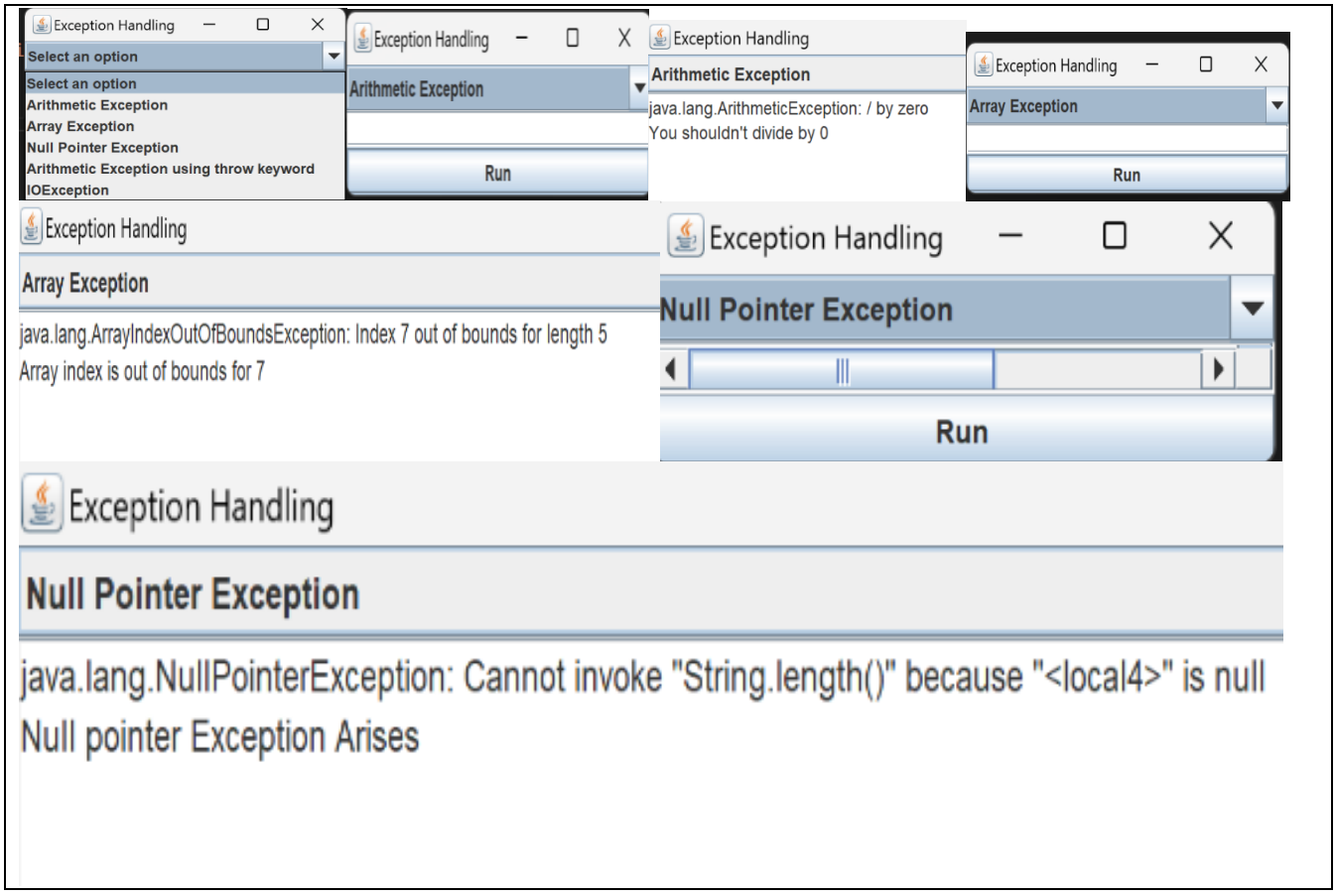
Code:

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

import java.awt.event.*;

public class gui4 extends JFrame implements ActionListener {

   private JComboBox comboBox;
   private JButton button;
   private JTextArea textArea;

   public gui4() {
      setTitle("Exception Handling");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());

      // Create a combo box to select an option
      String[] options = {"Select an option", "Arithmetic Exception", "Array Exception",
                  "Null Pointer Exception", "Arithmetic Exception using throw keyword", "IOException"};
```

```
        comboBox = new JComboBox<>(options);
        add(comboBox, BorderLayout.NORTH);

        // Create a button to run the selected option
        button = new JButton("Run");
        button.addActionListener(this);
        add(button, BorderLayout.SOUTH);

        // Create a text area to display the result
        textArea = new JTextArea();
        add(new JScrollPane(textArea), BorderLayout.CENTER);

        pack();
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        int choice = comboBox.getSelectedIndex();

        switch (choice) {
            case 0:
                textArea.setText("Please select an option");
                break;

            case 1:
                try {
                    int data = 10 / 0;
                } catch (ArithmeticException ex) {
                    textArea.setText(ex.toString() + "\nYou shouldn't divide by 0");
                }
                break;

            case 2:
                int[] arr = new int[5];

                try {
                    System.out.println(arr[7]);
                } catch (ArrayIndexOutOfBoundsException ex) {
                    textArea.setText(ex.toString() + "\nArray index is out of bounds for 7");
                }
                break;

            case 3:
                String s = null;
                try {
                    System.out.println(s.length());
                } catch (NullPointerException ex) {
                    textArea.setText(ex.toString() + "\nNull pointer Exception Arises");
                }
                break;

            case 4:
                int age = Integer.parseInt(JOptionPane.showInputDialog(this, "Enter Your Age"));
                if (age < 18) {
```

```
                //throw Arithmetic exception if not eligible to vote
                throw new ArithmeticException("Person is not eligible to vote");
            } else {
                textArea.setText("Person is eligible to vote!!");
            }
            break;

        default:
            textArea.setText("Wrong Choice");
            break;
    }
  }

  public static void main(String[] args) {
    new gui4();
  }
}
```

Screenshots/Results:



Assignment No. 5

Name of Student: Kartik Banshi Katkar

Batch: 2 **Branch: IT** **Roll No: 29**

Problem Statement

A fashion E-commerce company keeps a track of all the orders using an **ArrayList** and a class Order. Implement class **Order** and retrieve and return the list of items present in all the orders. Implement the logic inside **getItems()** method.

Test the functionalities using the **main()** method of the **Tester** class.

Sample Input and Output

Sample Input	Expected Output
orders=[Order(101,itemNames=[Jeans, Shirt, Belt],true), Order(102,itemNames=[Tie,Shirt],true),Order(103,itemNames=[Tshirt,Socks,Tie],true)	[Jeans,Shirt,Belt,Tie,Shirt,Tshirt,Socks,Tie]
orders=[Order(311,itemNames=[Sportswear, Dumbbell],true), Order(102,itemNames=[, Jeans],true),Order(103,itemNames=[Smartwatch,Fitnessband,Joggers],true)	Sportswear,Dumbbell,Smartwatch,Fitnessband,Joggers]

Code:

Order 1 code

```
import java.util.ArrayList;
import java.util.List;


class Order{
   private int orderId;
   private List itemNames;
   private boolean cashOnDelivary;
   public int getOrderId() {
      return orderId;
   }
   public void setOrderId(int orderId) {
      this.orderId = orderId;
   }
   public List getItemNames() {
      return itemNames;
   }
```

```
   public void setItemNames(List itemNames) {
      this.itemNames = itemNames;
   }
   public boolean getCashOnDelivary(){
      return cashOnDelivary;
   }
   public void setCashOnDelivary(boolean cashOnDelivary) {
      this.cashOnDelivary = cashOnDelivary;
   }
   public Order(int orderId, List itemNames, boolean cashOnDelivary ){
      this.orderId = orderId;
      this.cashOnDelivary = cashOnDelivary;
      this.itemNames = itemNames;

   }
}
public class Tester {
   public static void main(String[] args) {
      List itemnames1 = new ArrayList();
      itemnames1.add("Jeans");
      itemnames1.add("Shirt");
      itemnames1.add("Belt");
      Order o1 = new Order(101, itemnames1, true);

      List itemnames2 = new ArrayList();
      itemnames2.add("Tie");
      itemnames2.add("Shirt");
      Order o2 = new Order(102, itemnames2, true);

      List itemnames3 = new ArrayList();
      itemnames3.add("TShirts");
      itemnames3.add("Socks");
      itemnames3.add("Tie");
      Order o3 = new Order(103, itemnames3, true);

      List orders = new ArrayList();
      orders.add(o1);
      orders.add(o2);
      orders.add(o3);

      List items = getItems(orders);
      System.out.println("List of Items");

      /*for (String item : items) {
         System.out.print(item+ " ");
      }*/
      System.out.println(items);
   }
   private static List getItems(List orders) {
      List items = new ArrayList();
      for (Order order : orders) {
         items.addAll(order.getItemNames());
      }
      System.out.println(items);
      return items;
```

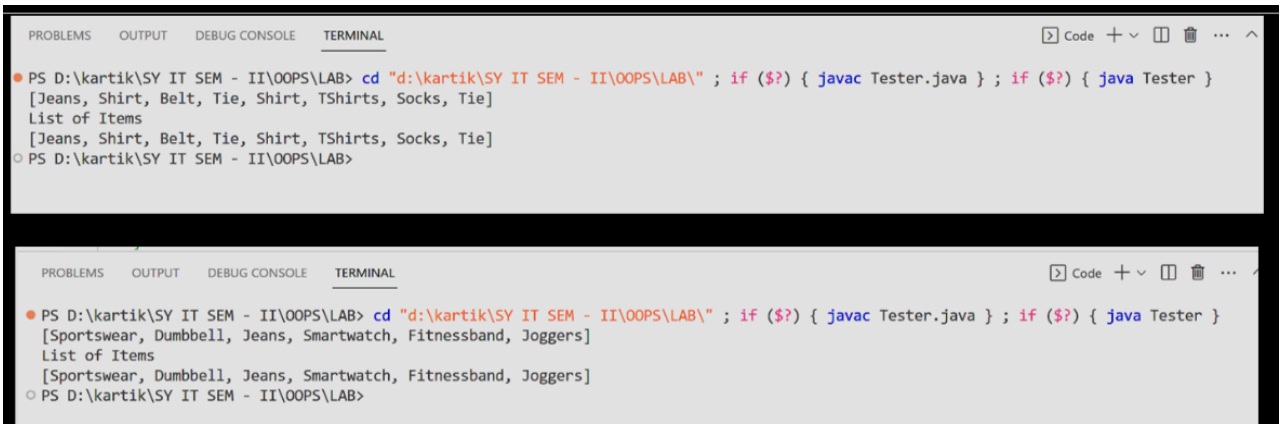
```
    }
}
```

Order 2 code

```
import java.util.ArrayList;
import java.util.List;


class Order1{
    private int orderId;
    private List itemNames;
    private boolean cashOnDelivary;
    public int getOrderId() {
        return orderId;
    }
    public void setOrderId(int orderId) {
        this.orderId = orderId;
    }
    public List getItemNames() {
        return itemNames;
    }
    public void setItemNames(List itemNames) {
        this.itemNames = itemNames;
    }
    public boolean getCashOnDelivary(){
        return cashOnDelivary;
    }
    public void setCashOnDelivary(boolean cashOnDelivary) {
        this.cashOnDelivary = cashOnDelivary;
    }
    public Order1(int orderId, List itemNames, boolean cashOnDelivary ){
        this.orderId = orderId;
        this.cashOnDelivary = cashOnDelivary;
        this.itemNames = itemNames;

    }
}
public class Tester {
    public static void main(String[] args) {
        List itemnames1 = new ArrayList();
        itemnames1.add("Sportswear");
        itemnames1.add("Dumbbell");

        Order1 o1 = new Order1(311, itemnames1, true);

        List itemnames2 = new ArrayList();
        itemnames2.add("Jeans");

        Order1 o2 = new Order1(102, itemnames2, true);

        List itemnames3 = new ArrayList();
        itemnames3.add("Smartwatch");
        itemnames3.add("Fitnessband");
        itemnames3.add("Joggers");
```

```
        Order1 o3 = new Order1(103, itemnames3, true);

        List orders = new ArrayList();
        orders.add(o1);
        orders.add(o2);
        orders.add(o3);

        List items = getItems(orders);
        System.out.println("List of Items");

        /*for (String item : items) {
            System.out.print(item+ " ");
        }*/
        System.out.println(items);
    }

    private static List getItems(List orders) {
        List items = new ArrayList();
        for (Order1 order : orders) {
            items.addAll(order.getItemNames());
        }
        System.out.println(items);
        return items;
    }
}
```

Screenshots/Results:



Assignment No. 6

Name of Student: Kartik Banshi Katkar

Batch: 2 **Branch: IT** **Roll No: 29**

Problem Statement
Implement various operations using JDBC Connectivity

Sample Input and Output

Sample Input/Parameter	Expected Output
Add Record	Dialog Message box to display Record Added Successfully
Update Record	Dialog Message box to display Record Updated Successfully
Delete Record	Dialog Message box to display Record Deleted Successfully
Display Record	Display Records in Tabular Format

Code:

```
import java.awt.*;


import javax.swing.*;
import javax.swing.table.*;

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

public class Demo_Sw {

	private JFrame frame;
	private JTextField text_1;
	private JTextField text_2;
	private JTextField text_3;
	private JTextField text_4;
	private JTextField text_5;

	/**
	 * Launch the application.
	 */
```

```
String driverName = "com.mysql.jdbc.Driver";
 static final String DB_URL="jdbc:mysql://localhost:3306/aids";
   static final String USER = "root";
   static final String PASS = "";

public static void main(String[] args) {
	EventQueue.invokeLater(new Runnable() {
		public void run() {
			try {
				Demo_Sw window = new Demo_Sw();
				window.frame.setVisible(true);
			} catch (Exception e) {
				e.printStackTrace();
			}
		}
	});
}

/**
 * Create the application.
 */
public Demo_Sw() {
	initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
	frame = new JFrame();
	frame.setBounds(100, 100, 450, 300);
	frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	frame.getContentPane().setLayout(null);

	JLabel Label_1 = new JLabel("Roll No.:");
	Label_1.setFont(new Font("Tahoma", Font.BOLD, 16));
	Label_1.setBounds(34, 11, 92, 30);
	frame.getContentPane().add(Label_1);

	JLabel Label_2 = new JLabel("Name:");
	Label_2.setFont(new Font("Tahoma", Font.BOLD, 16));
	Label_2.setBounds(34, 52, 92, 30);
	frame.getContentPane().add(Label_2);

	JLabel Label_3 = new JLabel("Class:");
	Label_3.setFont(new Font("Tahoma", Font.BOLD, 16));
	Label_3.setBounds(34, 93, 92, 30);
	frame.getContentPane().add(Label_3);

	JLabel Label_4 = new JLabel("Division:");
```

```
                Label_4.setFont(new Font("Tahoma", Font.BOLD, 16));
                Label_4.setBounds(34, 139, 92, 30);
                frame.getContentPane().add(Label_4);

                JLabel Label_5 = new JLabel("Semester:");
                Label_5.setFont(new Font("Tahoma", Font.BOLD, 16));
                Label_5.setBounds(34, 180, 92, 30);
                frame.getContentPane().add(Label_5);

                text_1 = new JTextField();
                text_1.setBounds(175, 17, 119, 23);
                frame.getContentPane().add(text_1);
                text_1.setColumns(10);

                text_2 = new JTextField();
                text_2.setColumns(10);
                text_2.setBounds(175, 59, 119, 23);
                frame.getContentPane().add(text_2);

                text_3 = new JTextField();
                text_3.setColumns(10);
                text_3.setBounds(175, 100, 119, 23);
                frame.getContentPane().add(text_3);

                text_4 = new JTextField();
                text_4.setColumns(10);
                text_4.setBounds(175, 146, 119, 23);
                frame.getContentPane().add(text_4);

                text_5 = new JTextField();
                text_5.setColumns(10);
                text_5.setBounds(175, 187, 119, 23);
                frame.getContentPane().add(text_5);

                JButton btn_1 = new JButton("INSERT");
                btn_1.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                String rollno = text_1.getText();
                  String name = text_2.getText();
                  String class1 = text_3.getText();
                  String division = text_4.getText();
                  String sem = text_5.getText();
  try {
                    Class.forName("com.mysql.jdbc.Driver");
                    Connection connection = DriverManager.getConnection(DB_URL,USER,PASS);
                    String query = "INSERT INTO stud_info values('" + rollno + "','" + name + "','" +
class1 + "','"+ division +"','"+sem+"')";
                    Statement sta = connection.createStatement();
                    sta.executeUpdate(query);
```

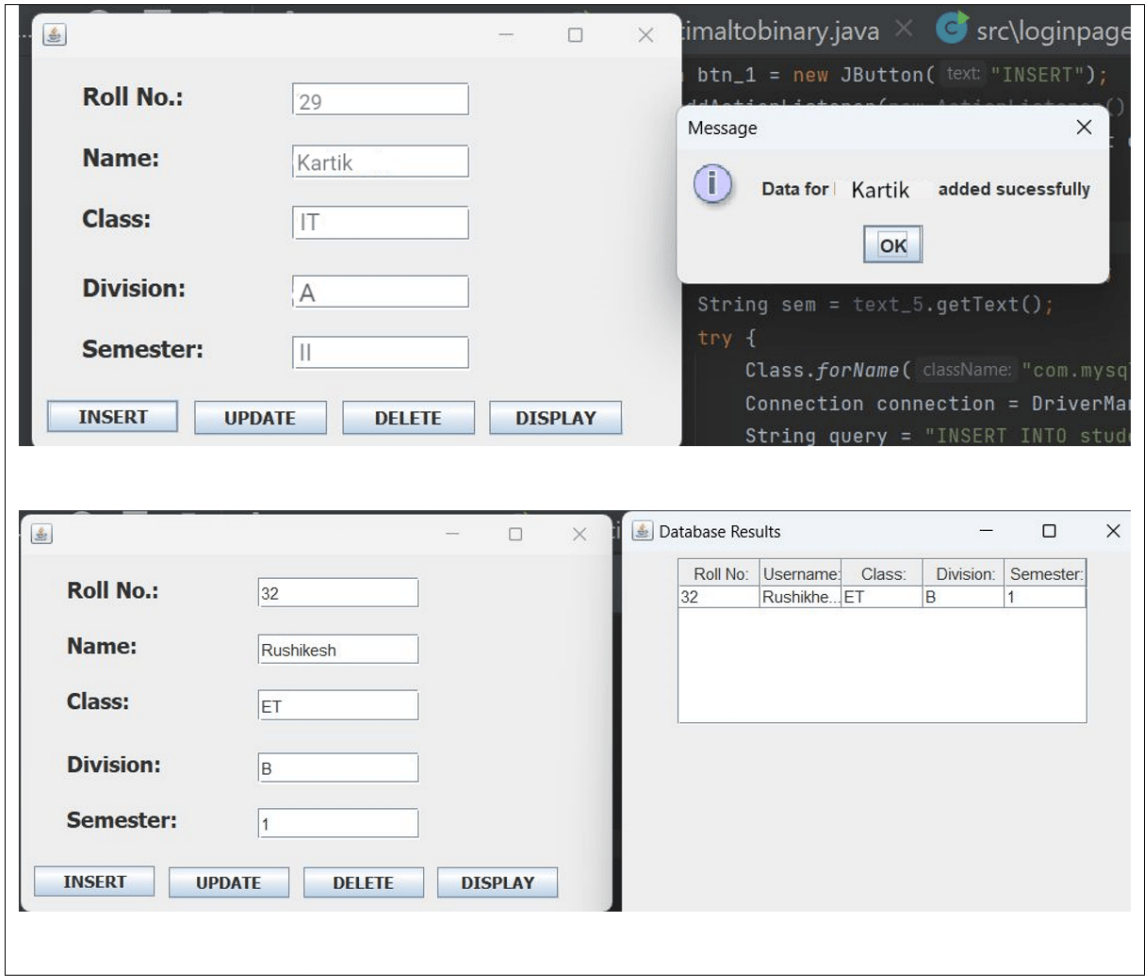
```
                    JOptionPane.showMessageDialog(btn_1,"Data for "+ name + " added sucessfully ");

                    connection.close();
                  } catch (Exception exception) {
                    exception.printStackTrace();
                  }

                        }
                });

                btn_1.setFont(new Font("Tahoma", Font.BOLD, 12));
                btn_1.setBounds(10, 229, 89, 23);
                frame.getContentPane().add(btn_1);

                JButton btn_2 = new JButton("UPDATE");
                btn_2.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                String rollno = text_1.getText();
            String name = text_2.getText();
            String class1 = text_3.getText();
            String division = text_4.getText();
            String sem = text_5.getText();

              try {
                                Class.forName("com.mysql.jdbc.Driver");
                                Connection connection =
DriverManager.getConnection(DB_URL,USER,PASS);
                                String query = "UPDATE stud_info set name="+"'"+name+"'"+"where
rollno=" + rollno;
                                Statement sta = connection.createStatement();
                                int x = sta.executeUpdate(query);
                                String query1 = "UPDATE stud_info set class1="+"'"+class1+"'"+"where
rollno=" + rollno;
                                Statement sta1 = connection.createStatement();
                                int x1 = sta1.executeUpdate(query1);
                                String query2 = "UPDATE stud_info set
division="+"'"+division+"'"+"where rollno=" + rollno;
                                Statement sta2 = connection.createStatement();
                                int x2 = sta2.executeUpdate(query2);
                                String query3 = "UPDATE stud_info set sem="+"'"+sem+"'"+"where
rollno=" + rollno;
                                Statement sta3 = connection.createStatement();
                                int x3 = sta3.executeUpdate(query3);
                                JOptionPane.showMessageDialog(btn_2,"Data for "+ name + "','"+ class1
+"','"+division+" and "+sem+" updated sucessfully ");

                                 connection.close();
                               } catch (Exception exception) {
                                 exception.printStackTrace();
                               }
```

```
                        }
                         });

                btn_2.setFont(new Font("Tahoma", Font.BOLD, 12));
                btn_2.setBounds(109, 230, 89, 23);
                frame.getContentPane().add(btn_2);

                JButton btn_3 = new JButton("DELETE");
                btn_3.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                String rollno = text_1.getText();
            String name = text_2.getText();
            String class1 = text_3.getText();
            String division = text_4.getText();
            String sem = text_5.getText();
                    try {
          Class.forName("com.mysql.jdbc.Driver");
          Connection connection = DriverManager.getConnection(DB_URL,USER,PASS);
          String query = "DELETE from stud_info where rollno=" + rollno;
          Statement sta = connection.createStatement();
          sta.executeUpdate(query);

          JOptionPane.showMessageDialog(btn_3,"Data for "+ name + " deleted sucessfully ");

           connection.close();
       } catch (Exception exception) {
          exception.printStackTrace();
       }
    }
});

                btn_3.setFont(new Font("Tahoma", Font.BOLD, 12));
                btn_3.setBounds(208, 230, 89, 23);
                frame.getContentPane().add(btn_3);

                JButton btn_4 = new JButton("DISPLAY");
                //setting the properties of second JFrame
      JFrame frame2 = new JFrame("Database Results");
      frame2.getContentPane().setLayout(new FlowLayout());
      frame2.setSize(400, 400);


      //Setting the properties of JTable and DefaultTableModel
      DefaultTableModel defaultTableModel = new DefaultTableModel();
      JTable table = new JTable(defaultTableModel);
      table.setPreferredScrollableViewportSize(new Dimension(300, 100));
      table.setFillsViewportHeight(true);
      frame2.getContentPane().add(new JScrollPane(table));
      defaultTableModel.addColumn("Roll No:");
      defaultTableModel.addColumn("Username:");
```

```
defaultTableModel.addColumn("Class:");
defaultTableModel.addColumn("Division:");
defaultTableModel.addColumn("Semester:");


            btn_4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                    int flag=0;


    try {
                Class.forName("com.mysql.jdbc.Driver");
                Connection connection = DriverManager.getConnection(DB_URL,USER,PASS);
                String query = "SELECT * from stud_info ";
                Statement sta = connection.createStatement();
                ResultSet rs= sta.executeQuery(query);


                while(rs.next())
                {

                                    String rollno=rs.getString(1);
                                    String name =rs.getString(2);
                                    String class1=rs.getString(3);
                                    String division=rs.getString(4);
                                    String sem=rs.getString(5);

                                    if (name.equalsIgnoreCase(rs.getString(2))) {
                        flag = 1;
                        defaultTableModel.addRow(new Object[]{rollno,name,
class1,division,sem});//Adding row in Table
                        frame2.setVisible(true);//Setting the visibility of second Frame
                        frame2.validate();
                        break;
                    }
                        if (flag == 0) {
                           JOptionPane.showMessageDialog(null, "No Such Username
Found");//When invalid username is entered
                        }

                   JOptionPane.showMessageDialog(btn_4,"Data for "+ name + " displayed successfully
");
                    }
                     connection.close();
                  } catch (Exception exception) {
                     exception.printStackTrace();
                  }
               }
                });
                btn_4.setFont(new Font("Tahoma", Font.BOLD, 12));
```

```
                btn_4.setBounds(307, 230, 89, 23);
                frame.getContentPane().add(btn_4);
        }
}
```

Screenshot/Result:



Assignment No. 7

Name of Student: Kartik Banshi Katkar

Batch: B2 **Branch: IT** **Roll No: 29**

Problem Statement:

Write a Java program that implements a multithreaded program has three threads. First thread generates a random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd the third thread will print the value of cube of the number.

THEORY:

The java run-time system depends on the threads for many things, and all the class libraries are designed with multithreading in mind. In fact, java uses threads to enable the entire environment to be asynchronous. This helps reduce inefficiency by preventing the waste of CPU cycles. The benefits of java's multithreading is that the main loop/polling mechanism is eliminated. One thread can pause without stopping other parts of your program. When a thread blocks in a java program, only the single thread that is blocked pauses. All other threads continue to run.

Program:

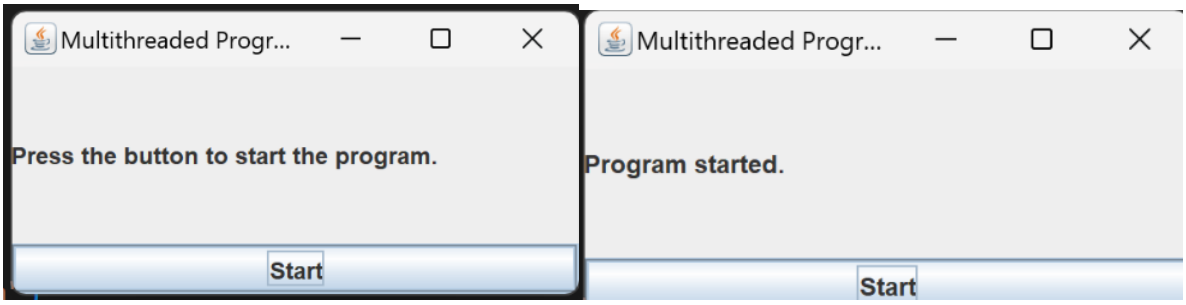
```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;
public class gui7 extends JFrame {
    private JLabel label;
    public gui7() {
        super("Multithreaded Program");
        label = new JLabel("Press the button to start the program.");
        add(label);
        JButton button = new JButton("Start");
        button.addActionListener(new ActionListener() {
```

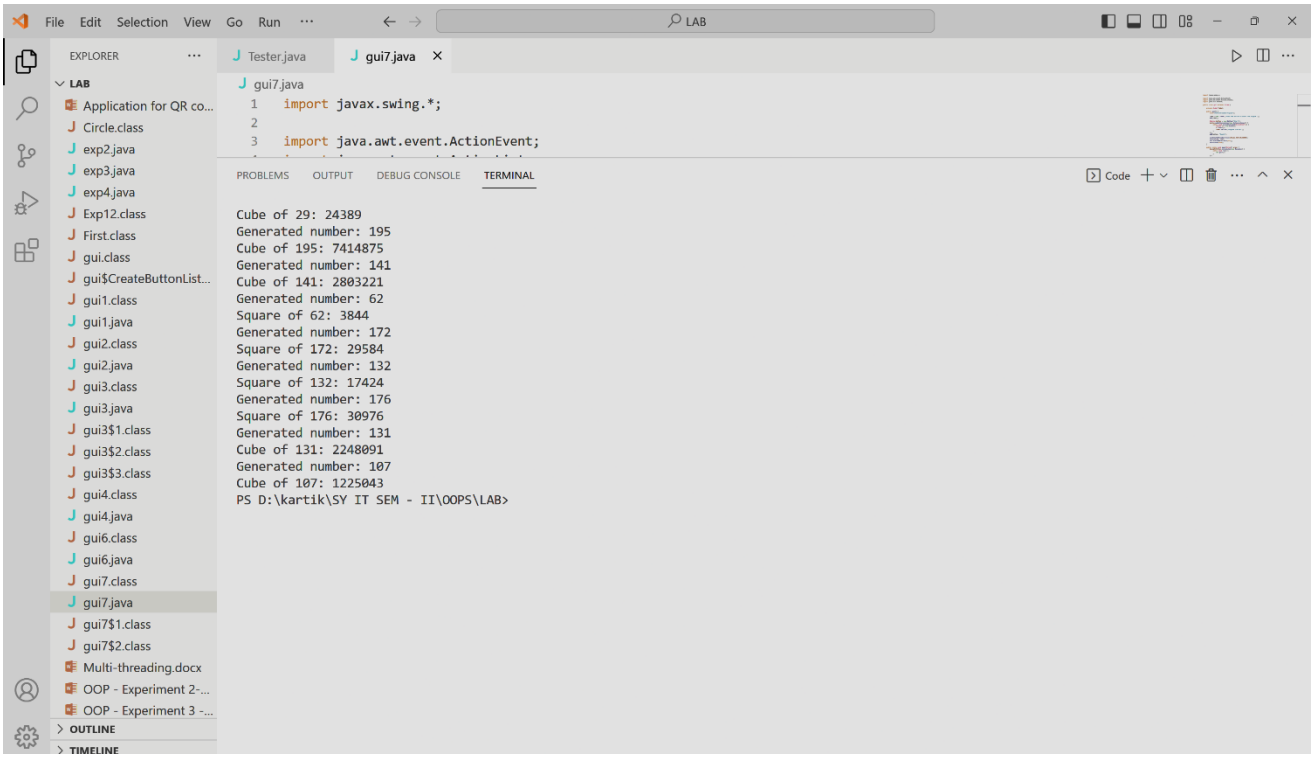
```
            public void actionPerformed(ActionEvent e) {
                Thread t1 = new First();
                t1.start();
                label.setText("Program started.");
            }
        });
        add(button, "South");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 150);
        setLocationRelativeTo(null);
        setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new gui7();
            }
        });
    }
}
class First extends Thread {
    public void run() {
        for (;;) {
            int r;
            Random d = new Random();
            r = d.nextInt(200) + 1;
            System.out.println("Generated number: " + r);
            if (r % 2 == 0) {
```

```
            Thread t2 = new Second(r);
            t2.start();
        } else {
            Thread t3 = new Third(r);
            t3.start();
        }
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {}
    }
  }
}
class Second extends Thread {
  private int number;

  public Second(int number) {
    this.number = number;
  }

  public void run() {
    int square = number * number;
    System.out.println("Square of " + number + ": " + square);
  }
}
class Third extends Thread {
  private int number;

  public Third(int number) {
```

```
        this.number = number;
    }
    public void run() {
        int cube = number * number * number;
        System.out.println("Cube of " + number + ": " + cube);
    }
}
```

Screenshots/Results:





Assignment No. 8

Name of Student: Kartik Banshi Katkar

Batch: B2 **Branch: IT** **Roll No: 29**

Aim:
Write a java program that handles all mouse events and shows the event name at the center of the window when mouse event is fired.

Theory:
To handle mouse events you must implement the MouseListener and the Mouse Motion Listener interfaces. These two interfaces contain methods that receive and process the various types of mouse events.

Program:

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

import javax.swing.*;

public class gui8 extends JFrame {

   private JLabel label;

   public gui8() {
      super("Mouse Event Handler");

      label = new JLabel("Welcome to Mouse Event Handler");
      label.setHorizontalAlignment(JLabel.CENTER);
      add(label);

      addMouseListener(new MouseAdapter() {
         public void mouseClicked(MouseEvent e) {
            label.setText("Mouse Clicked");
         }

         public void mouseEntered(MouseEvent e) {
            label.setText("Mouse Entered");
         }
```

```
            public void mouseExited(MouseEvent e) {
                label.setText("Mouse Exited");
            }
        });

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 150);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new gui8();
            }
        });
    }
}
```

Screenshots and results: