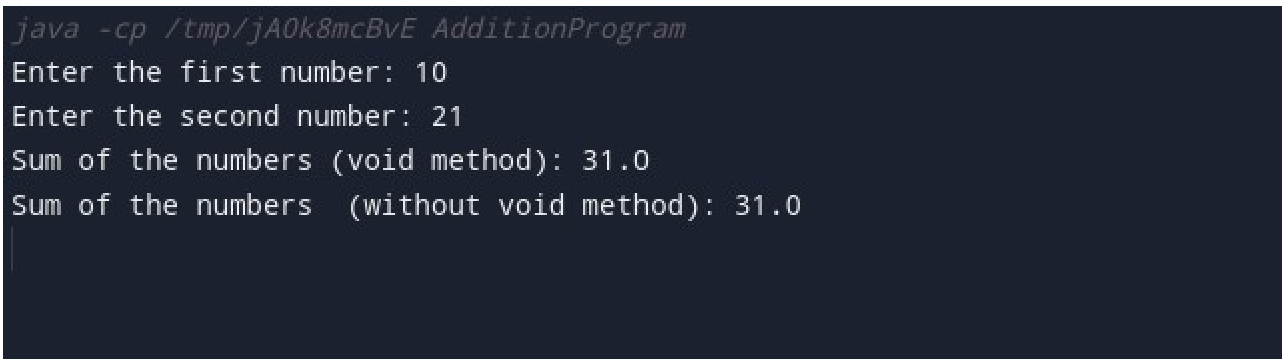
1. Write a program of class addition of two numbers by using void method and without void method.

```
import java.util.Scanner;

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

        System.out.print("Enter the first number: ");
        double num1 = scanner.nextDouble();

        System.out.print("Enter the second number: ");
        double num2 = scanner.nextDouble();

        // Call method with void
        addNumbersWithVoid(num1, num2);

        // Call method without void
        double sum = addNumbersWithoutVoid(num1, num2);
        System.out.println("Sum of the numbers  (without void method): " + sum);
    }

    // Void method for addition
    public static void addNumbersWithVoid(double a, double b) {
        double sum = a + b;
        System.out.println("Sum of the numbers (void method): " + sum);
    }

    // Method that returns the sum
    public static double addNumbersWithoutVoid(double a, double b) {
        return a + b;
    }
}
```

Output

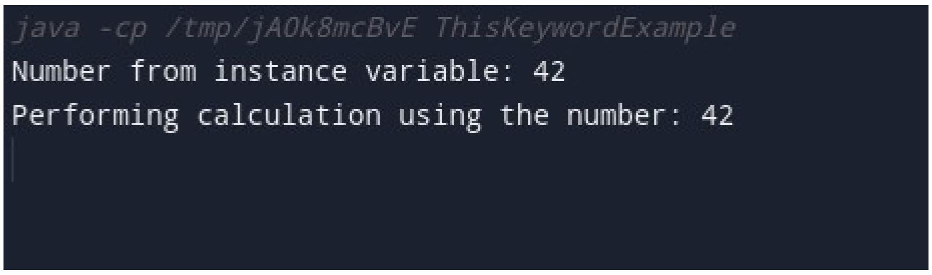


```
java -cp /tmp/jAOk8mcBvE AdditionProgram
Enter the first number: 10
Enter the second number: 21
Sum of the numbers (void method): 31.0
Sum of the numbers  (without void method): 31.0
```

2. Write a program of ‘This’ keyword can be used to refer current class instance variable and Method.

```
public class ThisKeywordExample {
    // Instance variable
    private int number;

    // Constructor
    public ThisKeywordExample(int number) {
        // Using 'this' to refer to the instance variable
        this.number = number;
    }

    // Method using 'this' to refer to instance method
    public void displayNumber() {
        System.out.println("Number from instance variable: " + this.number);
        this.performCalculation(); // Calling another method using 'this'
    }

    // Another method
    private void performCalculation() {
        System.out.println("Performing calculation using the number: " + this.number);
    }

    public static void main(String[] args) {
        // Creating an instance of the class
        ThisKeywordExample example = new ThisKeywordExample(42);

        // Calling the method using 'this'
        example.displayNumber();
    }
}
```

Output :



```
java -cp /tmp/jAOk8mcBvE ThisKeywordExample
Number from instance variable: 42
Performing calculation using the number: 42
```

3. Create a Package package1 having one class A contains msg method. Create second package mypack with class B, call msg method of A class in the second package.

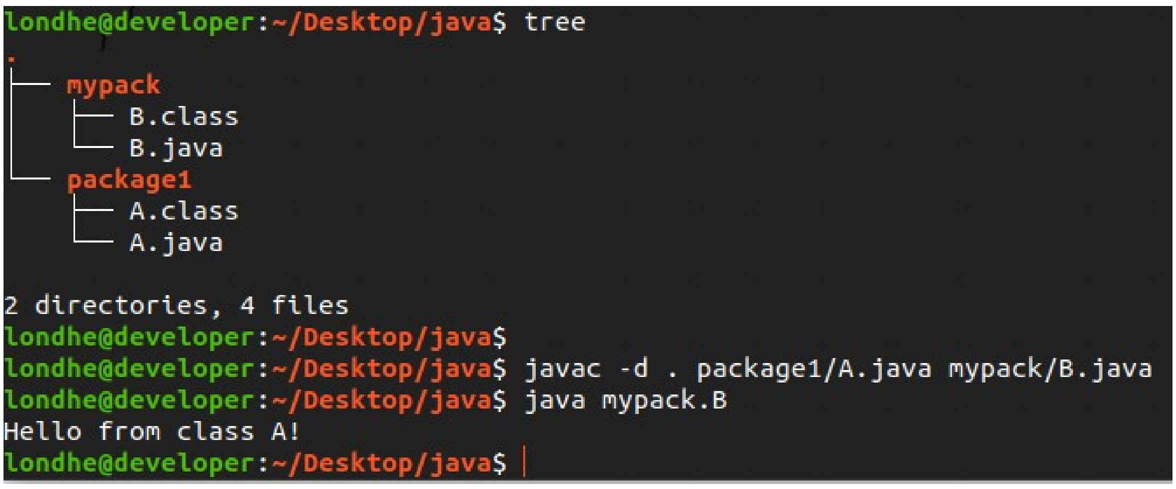
A.java (in package1):

```
package package1;

public class A {
    public void msg() {
        System.out.println("Hello from class A!");
    }
}
```

B.java (in mypack):

```
package mypack;
import package1.A; // Importing class A from package1

public class B {
    public static void main(String[] args) {
        // Creating an instance of class A
        A aObject = new A();

        // Calling the msg method from class A
        aObject.msg();
    }
}
```

Output :

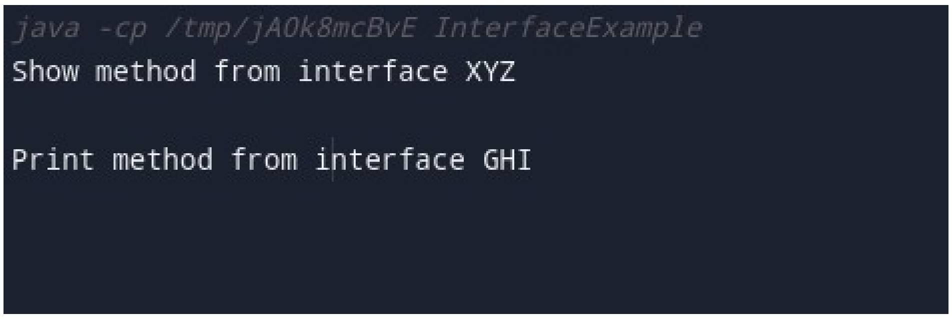


```
londhe@developer:~/Desktop/java$ tree
.
├── mypack
│   ├── B.class
│   └── B.java
└── package1
    ├── A.class
    └── A.java

2 directories, 4 files
londhe@developer:~/Desktop/java$
londhe@developer:~/Desktop/java$ javac -d . package1/A.java mypack/B.java
londhe@developer:~/Desktop/java$ java mypack.B
Hello from class A!
londhe@developer:~/Desktop/java$
```

4. Write a Java program of create Interface GHI having print method which extends another Interface XYZ having a show method . (create Interface extends another Interface program).

```
// Interface XYZ
interface XYZ {
   void show();
}

// Interface GHI extending XYZ
interface GHI extends XYZ {
   void print();
}

// Class implementing GHI
class MyClass implements GHI {
   @Override
   public void show() {
      System.out.println("Show method from interface XYZ");
   }

   @Override
   public void print() {
      System.out.println("Print method from interface GHI");
   }
}

// Main class
class InterfaceExample {
   public static void main(String[] args) {
      // Creating an instance of MyClass
      MyClass obj = new MyClass();

      // Calling methods from interfaces
      obj.show();
      System.out.println("");
      obj.print();
   }
}
```

Output :

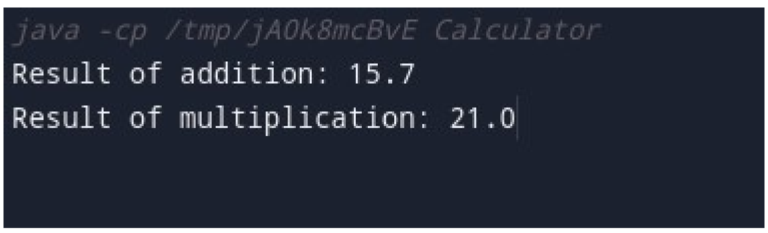


```
java -cp /tmp/jAOk8mcBvE InterfaceExample
Show method from interface XYZ

Print method from interface GHI
```

5. Create Constructor with accepting two numbers and perform any two types of constructor.

```
public class Calculator {
    private double result;

    // Constructor for addition
    public Calculator(double num1, double num2, String operation) {
        if ("add".equalsIgnoreCase(operation)) {
            this.result = num1 + num2;
        } else {
            System.out.println("Invalid operation. Defaulting to addition.");
            this.result = num1 + num2;
        }
    }

    // Constructor for multiplication
    public Calculator(double num1, double num2, boolean multiply) {
        if (multiply) {
            this.result = num1 * num2;
        } else {
            System.out.println("Invalid operation. Defaulting to multiplication.");
            this.result = num1 * num2;
        }
    }

    public double getResult() {
        return result;
    }

    public static void main(String[] args) {
        // Constructor with addition operation
        Calculator addCalculator = new Calculator(10.5, 5.2, "add");
        System.out.println("Result of addition: " + addCalculator.getResult());

        // Constructor with multiplication operation
        Calculator multiplyCalculator = new Calculator(7.0, 3.0, true);
        System.out.println("Result of multiplication: " + multiplyCalculator.getResult());
    }
}
```

Output :

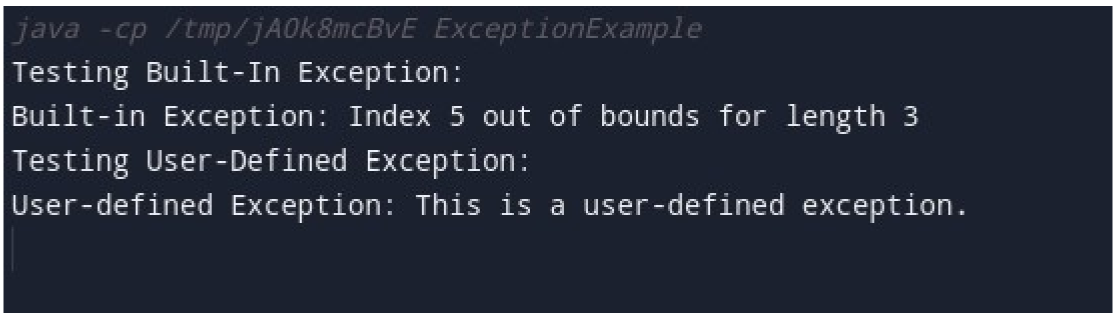


```
java -cp /tmp/jAOk8mcBvE Calculator
Result of addition: 15.7
Result of multiplication: 21.0
```

6. Write a Java program to test any one Build in exception and one user defined exception.

```
// User-defined exception
class CustomException extends Exception {
    public CustomException(String message) {
        super(message);
    }
}

class ExceptionExample {
    // Method to test built-in exception
    public static void testBuiltInException() {
        try {
            int[] numbers = {1, 2, 3};
            // Trying to access an index that is out of bounds
            int value = numbers[5];
            System.out.println("Value: " + value);  // This line will not be reached
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Built-in Exception: " + e.getMessage());
        }
    }

    // Method to test user-defined exception
    public static void testUserDefinedException() {
        try {
            throw new CustomException("This is a user-defined exception.");
        } catch (CustomException e) {
            System.out.println("User-defined Exception: " + e.getMessage());
        }
    }

    public static void main(String[] args) {
        // Test the built-in exception
        System.out.println("Testing Built-In Exception:");
        testBuiltInException();

        // Test the user-defined exception
        System.out.println("\nTesting User-Defined Exception:");
        testUserDefinedException();
    }
}
```

Output :

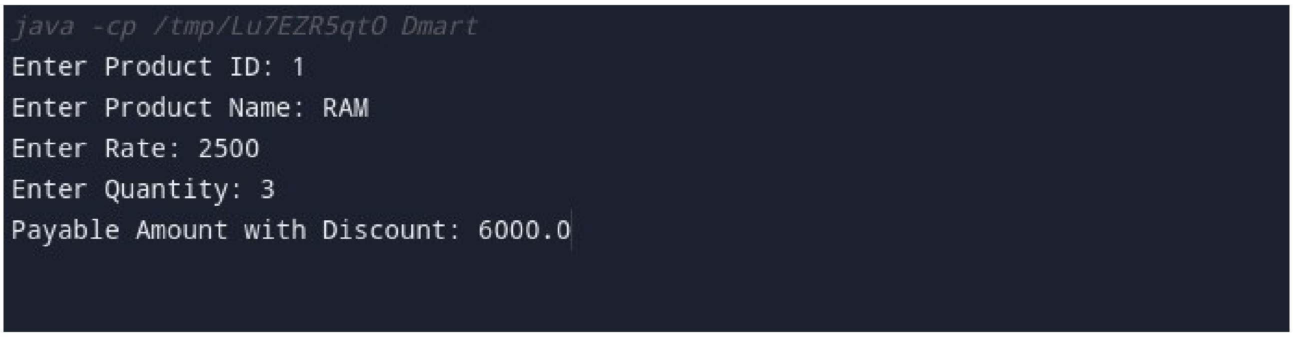


```
java -cp /tmp/jAOk8mcBvE ExceptionExample
Testing Built-In Exception:
Built-in Exception: Index 5 out of bounds for length 3
Testing User-Defined Exception:
User-defined Exception: This is a user-defined exception.
```

7. Write a program to create dmart class with accepting values of product id,name,rate and quantity and calculate payable amt with discount If amt 5000 3000 =20 %, 3000-2000 =15%, 2000-1000=10% , Otherwise no discount.

```
import java.util.Scanner;

class Dmart {
    private int productId;
    private String productName;
    private double rate;
    private int quantity;

    // Constructor to initialize values
    public Dmart(int productId, String productName, double rate, int quantity) {
        this.productId = productId;
        this.productName = productName;
        this.rate = rate;
        this.quantity = quantity;
    }

    // Method to calculate payable amount with discount
    public double calculatePayableAmount() {
        double totalAmount = rate * quantity;
        double discountRate = 0.0;

        // Apply discount based on totalAmount
        if (totalAmount >= 5000) {
            discountRate = 0.20;
        } else if (totalAmount >= 3000) {
            discountRate = 0.15;
        } else if (totalAmount >= 2000) {
            discountRate = 0.10;
        }

        double discount = totalAmount * discountRate;
        double payableAmount = totalAmount - discount;

        return payableAmount;
    }

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

        System.out.print("Enter Product ID: ");
        int productId = scanner.nextInt();

        System.out.print("Enter Product Name: ");
        String productName = scanner.next();
        System.out.print("Enter Rate: ");
        double rate = scanner.nextDouble();

        System.out.print("Enter Quantity: ");
        int quantity = scanner.nextInt();

        // Create an instance of Dmart
        Dmart dmart = new Dmart(productId, productName, rate, quantity);

        // Calculate and display payable amount with discount
        double payableAmount = dmart.calculatePayableAmount();
        System.out.println("Payable Amount with Discount: " + payableAmount);
    }
}
```

Output :



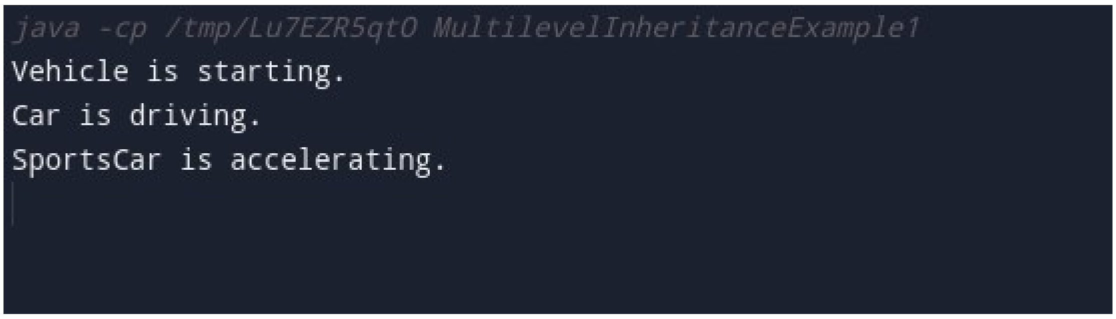
```
java -cp /tmp/Lu7EZR5qtO Dmart
Enter Product ID: 1
Enter Product Name: RAM
Enter Rate: 2500
Enter Quantity: 3
Payable Amount with Discount: 6000.0
```

8. Write any one example of Multilevel inheritance in JAVA.

Example-1.java

```
// Base class
class Vehicle {
    void start() {
        System.out.println("Vehicle is starting.");
    }
}

// Derived class inheriting from Vehicle
class Car extends Vehicle {
    void drive() {
        System.out.println("Car is driving.");
    }
}

// Another class inheriting from Car (Multilevel Inheritance)
class SportsCar extends Car {
    void accelerate() {
        System.out.println("SportsCar is accelerating.");
    }
}

// Main class to test multilevel inheritance
class MultilevelInheritanceExample1 {
    public static void main(String[] args) {
        // Create an instance of the SportsCar class
        SportsCar mySportsCar = new SportsCar();

        // Access methods from different levels of inheritance
        mySportsCar.start();      // Method from Vehicle class
        mySportsCar.drive();      // Method from Car class
        mySportsCar.accelerate(); // Method from SportsCar class
    }
}
```

Output :

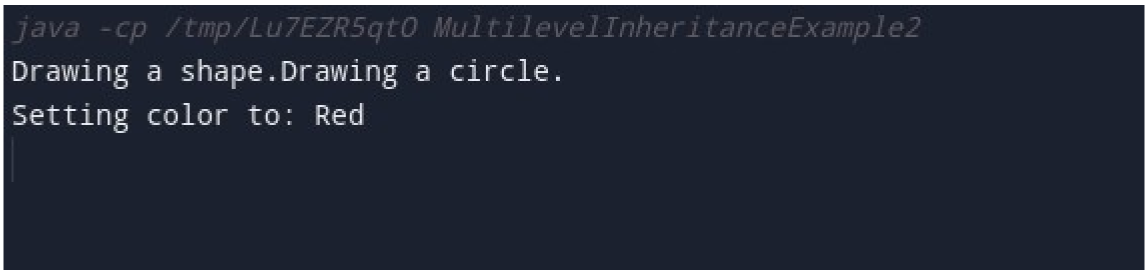


```
java -cp /tmp/Lu7EZR5qtO MultilevelInheritanceExample1
Vehicle is starting.
Car is driving.
SportsCar is accelerating.
```

Example-2.java

```java
// Base class
class Shape {
    void draw() {
        System.out.println("Drawing a shape.");
    }
}

// Derived class inheriting from Shape
class Circle extends Shape {
    void drawCircle() {
        System.out.println("Drawing a circle.");
    }
}

// Another class inheriting from Circle (Multilevel Inheritance)
class ColoredCircle extends Circle {
    void setColor(String color) {
        System.out.println("Setting color to: " + color);
    }
}

// Main class to test multilevel inheritance
class MultilevelInheritanceExample2 {
    public static void main(String[] args) {
        // Create an instance of the ColoredCircle class
        ColoredCircle myColoredCircle = new ColoredCircle();

        // Access methods from different levels of inheritance
        myColoredCircle.draw();       // Method from Shape class
        myColoredCircle.drawCircle(); // Method from Circle class
        myColoredCircle.setColor("Red"); // Method from ColoredCircle class
    }
}
```

Output :

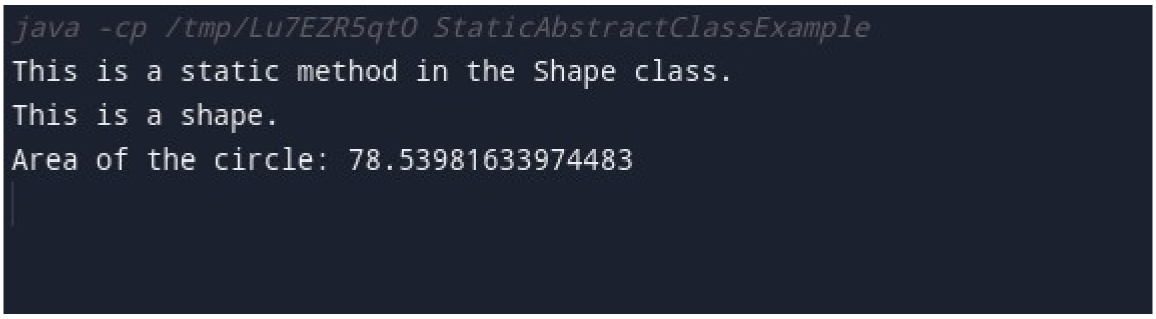


```
java -cp /tmp/Lu7EZR5qtO MultilevelInheritanceExample2
Drawing a shape.Drawing a circle.
Setting color to: Red
```

9. WAP of static, abstract class keyword in JAVA.

```
// Abstract class with abstract and non-abstract methods
abstract class Shape {
    // Abstract method (to be implemented by subclasses)
    abstract double calculateArea();
    // Non-abstract method
    void display() {
        System.out.println("This is a shape.");
    }
    // Static method
    static void staticMethod() {
        System.out.println("This is a static method in the Shape class.");
    }
}
// Concrete subclass implementing the abstract class
class Circle extends Shape {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }
    // Implementation of abstract method from the superclass
    @Override
    double calculateArea() {
        return Math.PI * radius * radius;
    }
}

// Main class to test static and abstract class
class StaticAbstractClassExample {
    public static void main(String[] args) {
        // Accessing static method without creating an instance
        Shape.staticMethod();

        // Creating an instance of the Circle class
        Circle circle = new Circle(5.0);

        // Calling methods from the abstract class
        circle.display(); // Non-abstract method
        double area = circle.calculateArea(); // Abstract method
        System.out.println("Area of the circle: " + area);
    }
}
```

Output :

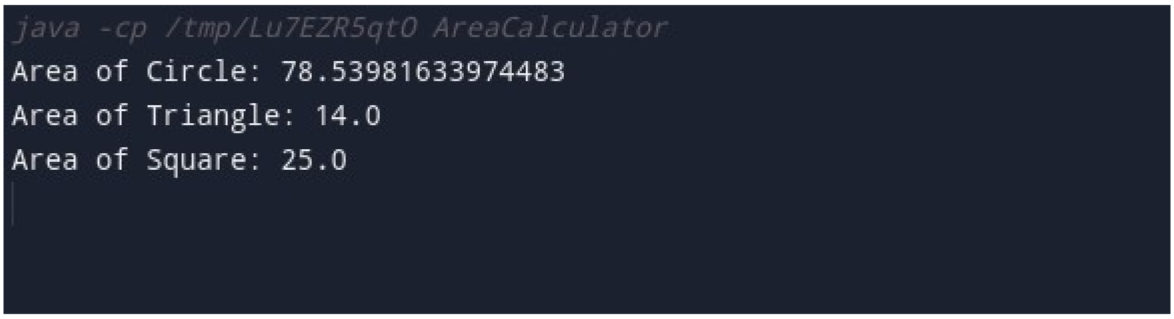


```
java -cp /tmp/Lu7EZR5qtO StaticAbstractClassExample
This is a static method in the Shape class.
This is a shape.
Area of the circle: 78.53981633974483
```

10. Calculate area of Circle, triangle and Square using Method Overloading.

```
class AreaCalculator {

    // Method to calculate the area of a circle
    public static double calculateArea(double radius) {
        return Math.PI * radius * radius;
    }

    // Method to calculate the area of a triangle
    public static double calculateArea(double base, double height) {
        return 0.5 * base * height;
    }

    // Method to calculate the area of a square
    public static double calculateArea(int sideLength) {
        return sideLength * sideLength;
    }

    public static void main(String[] args) {
        // Calculate and display the area of a circle
        double circleArea = calculateArea(5.0);
        System.out.println("Area of Circle: " + circleArea);

        // Calculate and display the area of a triangle
        double triangleArea = calculateArea(4.0, 7.0);
        System.out.println("Area of Triangle: " + triangleArea);

        // Calculate and display the area of a square
        double squareArea = calculateArea(5);
        System.out.println("Area of Square: " + squareArea);
    }
}
```

Output :

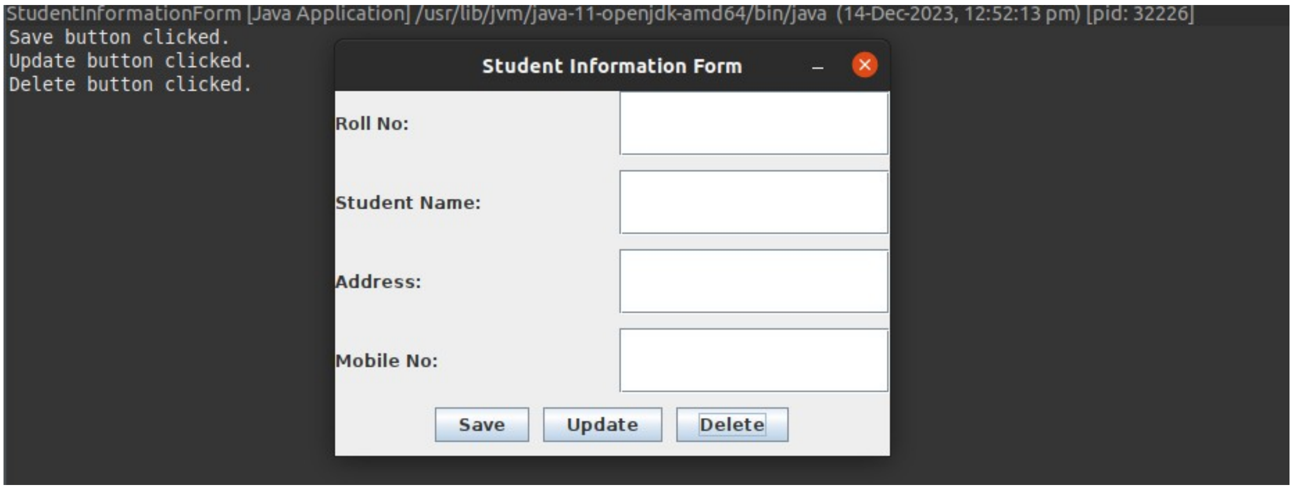


```
java -cp /tmp/Lu7EZR5qtO AreaCalculator
Area of Circle: 78.53981633974483
Area of Triangle: 14.0
Area of Square: 25.0
```

11. Create student information form using Java AWT components (Accept Rollno, Student Name, Address, Mob no) with Save, Update and delete components.

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

public class StudentInformationForm extends Frame {
    private Label lblRollNo, lblName, lblAddress, lblMobNo;
    private TextField tfRollNo, tfName, tfAddress, tfMobNo;
    private Button btnSave, btnUpdate, btnDelete;

    public StudentInformationForm() {
        setTitle("Student Information Form");
        setSize(400, 300);
        setLayout(new GridLayout(5, 2));
        setLocationRelativeTo(null); // Center the frame
        setResizable(false);

        // Labels
        lblRollNo = new Label("Roll No:");
        lblName = new Label("Student Name:");
        lblAddress = new Label("Address:");
        lblMobNo = new Label("Mobile No:");

        // TextFields
        tfRollNo = new TextField();
        tfName = new TextField();
        tfAddress = new TextField();
        tfMobNo = new TextField();

        // Buttons
        btnSave = new Button("Save");
        btnUpdate = new Button("Update");
        btnDelete = new Button("Delete");

        // Adding components to the frame
        add(lblRollNo);
        add(tfRollNo); add(lblName);
        add(tfName);
        add(lblAddress);
        add(tfAddress);
        add(lblMobNo);
        add(tfMobNo);
        add(btnSave);
        add(btnUpdate);
        add(btnDelete);

        // Button actions
        btnSave.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                saveStudentInformation();
            }
        });

        btnUpdate.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                updateStudentInformation();
            }
        });

        btnDelete.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                deleteStudentInformation();
            }
        });

        // Closing the window
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent windowEvent) {
                System.exit(0);
            }
        });

        setVisible(true);
    }

    private void saveStudentInformation() {
        // Implement the logic for saving student information
        System.out.println("Save button clicked.");
    }

    private void updateStudentInformation() {
        // Implement the logic for updating student information
        System.out.println("Update button clicked.");
    }

    private void deleteStudentInformation() {
        // Implement the logic for deleting student information
        System.out.println("Delete button clicked.");
    }

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

Output :



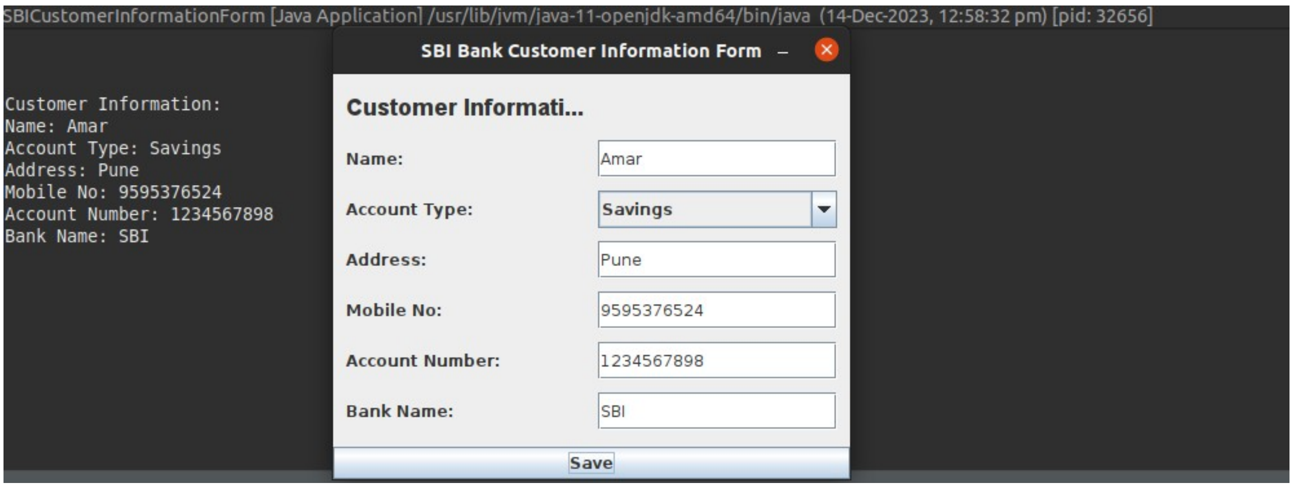
```
StudentInformationForm [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java  (14-Dec-2023, 12:52:13 pm) [pid: 32226]
Save button clicked.
Update button clicked.
Delete button clicked.
```

12. Create SBI Bank customer information form using Java Swing components

```
package src;

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

public class SBICustomerInformationForm extends JFrame {
   private static final long serialVersionUID = 1L;
         private JLabel lblTitle, lblName, lblAccountType, lblAddress, lblMobileNo,
lblAccountNumber, lblBankName;
   private JTextField tfName, tfAddress, tfMobileNo, tfAccountNumber, tfBankName;
   private JComboBox<String> cbAccountType;
   private JButton btnSave;

   public SBICustomerInformationForm() {
      setTitle("SBI Bank Customer Information Form");
      setSize(400, 350);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLocationRelativeTo(null); // Center the frame

      // Title Label
      lblTitle = new JLabel("Customer Information");
      lblTitle.setFont(new Font("Arial", Font.BOLD, 18));

      // Labels
      lblName = new JLabel("Name:");
      lblAccountType = new JLabel("Account Type:");
      lblAddress = new JLabel("Address:");
      lblMobileNo = new JLabel("Mobile No:");
      lblAccountNumber = new JLabel("Account Number:");
      lblBankName = new JLabel("Bank Name:");

      // TextFields
      tfName = new JTextField();
      tfAddress = new JTextField();
      tfMobileNo = new JTextField();
      tfAccountNumber = new JTextField();
      tfBankName = new JTextField();

      // ComboBox for Account Type
      String[] accountTypes = {"Savings", "Current", "Fixed Deposit"};
      cbAccountType = new JComboBox<>(accountTypes);
```

```
        // Button
        btnSave = new JButton("Save");

        // Layout
        JPanel panel = new JPanel(new GridLayout(7, 2, 10, 10));
        panel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        panel.add(lblTitle);
        panel.add(new JLabel()); // Empty label for spacing
        panel.add(lblName);
        panel.add(tfName);
        panel.add(lblAccountType);
        panel.add(cbAccountType);
        panel.add(lblAddress);
        panel.add(tfAddress);
        panel.add(lblMobileNo);
        panel.add(tfMobileNo);
        panel.add(lblAccountNumber);
        panel.add(tfAccountNumber);
        panel.add(lblBankName);
        panel.add(tfBankName);

        // Adding components to the frame
        add(panel, BorderLayout.CENTER);
        add(btnSave, BorderLayout.SOUTH);

        // Button actions
        btnSave.addActionListener(e -> saveCustomerInformation());

        // Set frame properties
        setResizable(false);
        setVisible(true);
    }

    private void saveCustomerInformation() {
        // Implement the logic for saving customer information
        String name = tfName.getText();
        String accountType = (String) cbAccountType.getSelectedItem();
        String address = tfAddress.getText();
        String mobileNo = tfMobileNo.getText();
        String accountNumber = tfAccountNumber.getText();
        String bankName = tfBankName.getText();

        // Print or save the customer information (you can modify this part)
        System.out.println("Customer Information:");
        System.out.println("Name: " + name);
        System.out.println("Account Type: " + accountType);
        System.out.println("Address: " + address);
        System.out.println("Mobile No: " + mobileNo);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Bank Name: " + bankName);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new SBICustomerInformationForm());
    }
}
```

Output :



```
SBICustomerInformationForm [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java  (14-Dec-2023, 12:58:32 pm) [pid: 32656]

Customer Information:
Name: Amar
Account Type: Savings
Address: Pune
Mobile No: 9595376524
Account Number: 1234567898
Bank Name: SBI
```

SBI Bank Customer Information Form

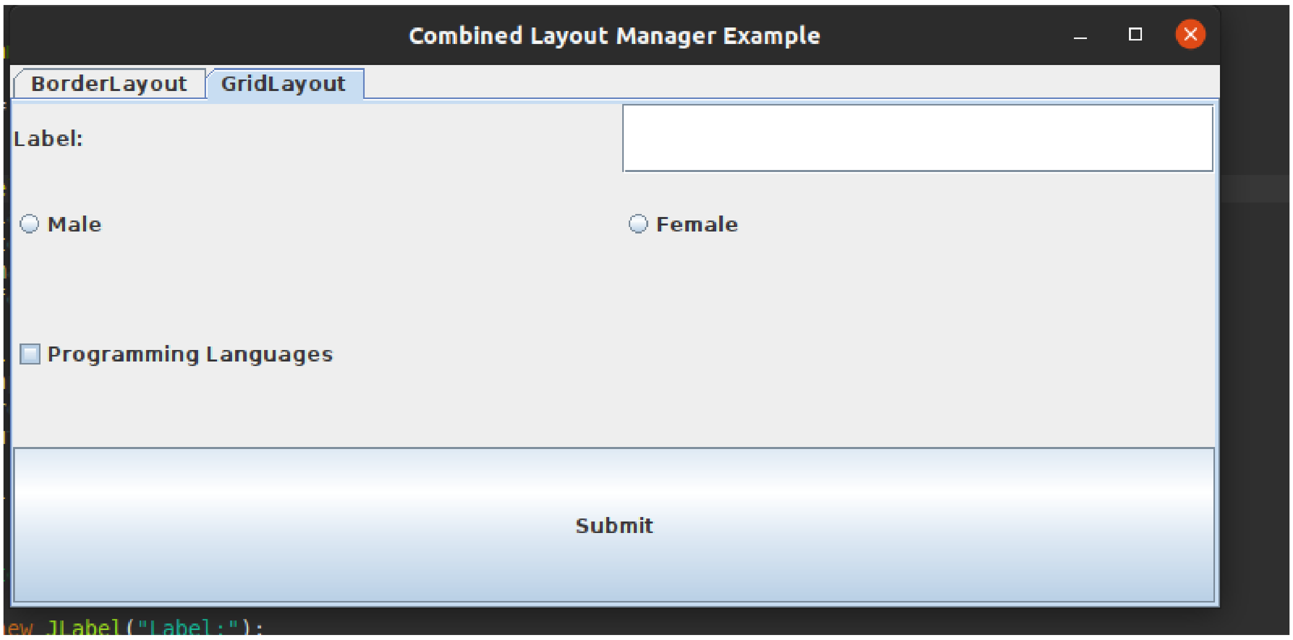
Customer Informati...

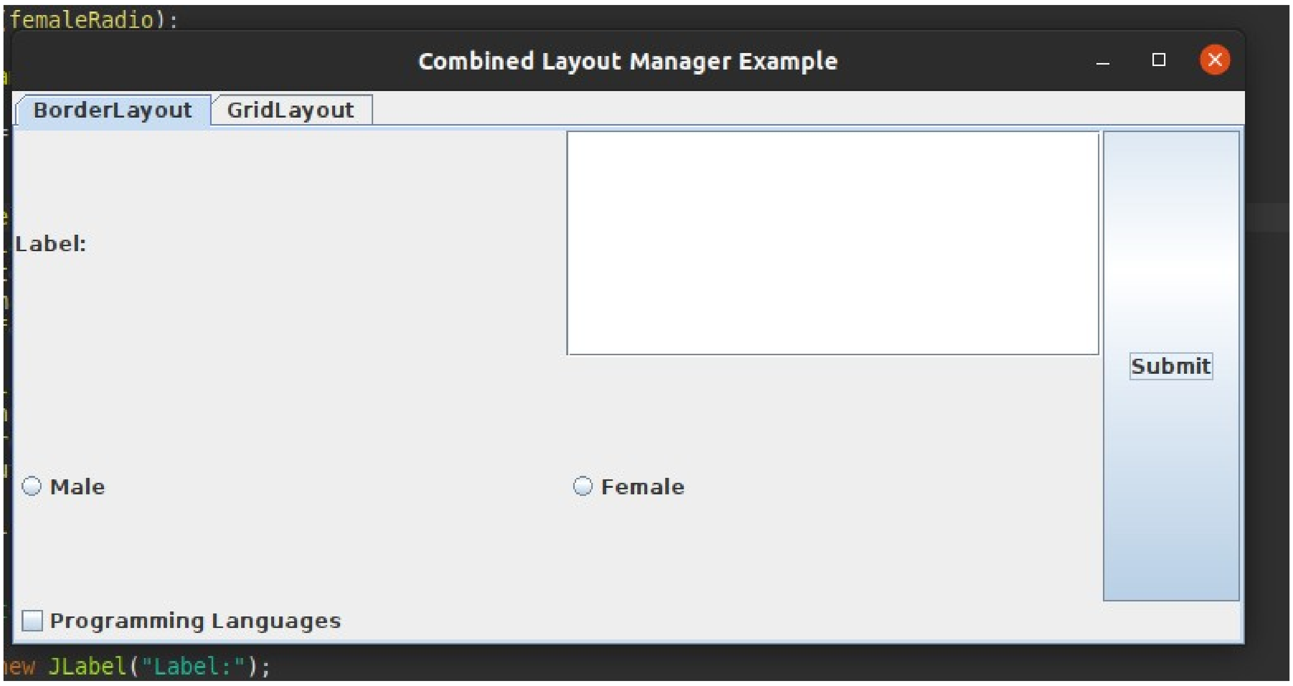
Name:	Amar
Account Type:	Savings
Address:	Pune
Mobile No:	9595376524
Account Number:	1234567898
Bank Name:	SBI

13. Write program of any two Layout Managers JAVA.

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

public class CombinedLayoutManagerExample extends JFrame {

  private static final long serialVersionUID = 1L;

      public CombinedLayoutManagerExample() {
    setTitle("Combined Layout Manager Example");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(500, 300);

    // Create tabs
    JTabbedPane tabbedPane = new JTabbedPane();

    // Add BorderLayout Example
    JPanel borderLayoutPanel = createBorderLayoutExample();
    tabbedPane.addTab("BorderLayout", borderLayoutPanel);

    // Add GridLayout Example
    JPanel gridLayoutPanel = createGridLayoutExample();
    tabbedPane.addTab("GridLayout", gridLayoutPanel);

    // Add tabs to the frame
    add(tabbedPane);

    // Set frame properties
    setLocationRelativeTo(null);
    setVisible(true);
  }

  private JPanel createBorderLayoutExample() {
    // Components
    JLabel label = new JLabel("Label:");
    JTextField textField = new JTextField();

    JRadioButton maleRadio = new JRadioButton("Male");
    JRadioButton femaleRadio = new JRadioButton("Female");
    ButtonGroup genderGroup = new ButtonGroup();
    genderGroup.add(maleRadio);
    genderGroup.add(femaleRadio);

    JCheckBox programmingCheckbox = new JCheckBox("Programming Languages");

    JButton button = new JButton("Submit");

    // Panels
}

        JPanel inputPanel = new JPanel(new GridLayout(2, 2, 10, 10));
        inputPanel.add(label);
        inputPanel.add(textField);
        inputPanel.add(maleRadio);
        inputPanel.add(femaleRadio);

        JPanel mainPanel = new JPanel(new BorderLayout());
        mainPanel.add(inputPanel, BorderLayout.CENTER);
        mainPanel.add(programmingCheckbox, BorderLayout.SOUTH);
        mainPanel.add(button, BorderLayout.EAST);

        return mainPanel;
    }

    private JPanel createGridLayoutExample() {
        // Components
        JLabel label = new JLabel("Label:");
        JTextField textField = new JTextField();

        JRadioButton maleRadio = new JRadioButton("Male");
        JRadioButton femaleRadio = new JRadioButton("Female");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleRadio);
        genderGroup.add(femaleRadio);

        JCheckBox programmingCheckbox = new JCheckBox("Programming Languages");

        JButton button = new JButton("Submit");

        // Panels
        JPanel inputPanel = new JPanel(new GridLayout(2, 2, 10, 10));
        inputPanel.add(label);
        inputPanel.add(textField);
        inputPanel.add(maleRadio);
        inputPanel.add(femaleRadio);

        JPanel mainPanel = new JPanel(new GridLayout(3, 1, 10, 10));
        mainPanel.add(inputPanel);
        mainPanel.add(programmingCheckbox);
        mainPanel.add(button);

        return mainPanel;
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(CombinedLayoutManagerExample::new);
    }
}
```

Output :


Combined Layout Manager Example
BorderLayout
GridLayout
Label:
Male
Female
Programming Languages
Submit
ew JLabel("Label:");


femaleRadio);
Combined Layout Manager Example
BorderLayout
GridLayout
Label:
Submit
Male
Female
Programming Languages
ew JLabel("Label:");