

**Name:** Mukesh Choudhary

**Student ID:** 590013672

**Branch:** MCA

**Batch:** B4

**Instructor:** Suvojit Dhara

**Date:** 20 October 2024

## Java Programming

### Assignment - 1.

Q.1

Sol To solve the problem, follow the steps:

#### Algorithm:

##### 1. Input two $4 \times 4$ matrices:

- The user will input two  $4 \times 4$  matrices (A and B)
- we'll take input for the matrices in a 2D array format.

##### 2. Matrix multiplication (AB and BA):

$$(AB)_{ij} = \sum_{k=1}^4 A_{ik} \times B_{kj}$$

- To multiply matrices A and B, we'll apply the rule of matrix multiplication.

- perform matrix multiplication for both AB and BA to compute two separate results.

- For BA, switch the matrices (multiply B by A instead of A by B).

##### 3. Verification:

- Compare the matrices from the two operations (AB and BA).

- Since matrix multiplication is generally non-commutative expect  $AB \neq BA$ . Print a message to verify whether they are equal or not.

#### 4. Output:

- Print matrices A, B, AB and BA.
- Print whether AB is equal to BA or not.

#### CODE

```
import java.util.Scanner;
public class MatrixMultiplication {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int [][] A = new int [4][4], B = new int [4][4], AB =
            new int [4][4], BA = new int [4][4];
        System.out.println ("Enter matrix A (4x4).:");
        for (int i=0; i<4; i++) for (int j=0; j<4; j++)
            A [i][j] = sc.nextInt ();
    }
}
```

```
System.out.println ("Enter matrix B (4x4).:");
for (int i=0; i<4; i++) for (int j=0; j<4; j++)
    B [i][j] = sc.nextInt ();
```

```
for (int i=0; i<4; i++) {
    for (int j=0; j<4; j++) {
        for (int k=0; k<4; k++) {
            AB [i][j] += A [i][k] * B [k][j];
            BA [i][j] += B [i][k] * A [k][j];
        }
    }
}
```

```
boolean isEqual = true;  
System.out.println("Matrix AB: ");  
for (int i=0; i<4; i++) {  
    for (int j=0; j<4; j++) {  
        System.out.print(AB[i][j] + " ");  
    }  
    if (AB[i][j] != BA[i][j]) isEqual = false;  
}  
System.out.println();  
}  
System.out.println("Matrix BA: ");  
for (int i=0; i<4; i++) {  
    for (int j=0; j<4; j++) {  
        System.out.print(BA[i][j] + " ");  
    }  
    System.out.println();  
}  
System.out.println(isEqual ? "AB = BA": "AB ≠ BA");  
sc.close();  
}
```

```
import java.util.Scanner;

public class MatrixMultiplication {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[][] A = new int[4][4], B = new int[4][4], AB = new int[4][4], BA = new int[4][4];

        // Input matrices A and B
        System.out.println("Enter Matrix A (4x4): ");
        for (int i = 0; i < 4; i++) for (int j = 0; j < 4; j++) A[i][j] = sc.nextInt();
        System.out.println("Enter Matrix B (4x4): ");
        for (int i = 0; i < 4; i++) for (int j = 0; j < 4; j++) B[i][j] = sc.nextInt();

        // Matrix multiplication for AB and BA
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                for (int k = 0; k < 4; k++) {
                    AB[i][j] += A[i][k] * B[k][j];
                    BA[i][j] += B[i][k] * A[k][j];
                }
            }
        }
    }
}
```

```
// Print results and check AB == BA
boolean isEqual = true;
System.out.println(x:"Matrix AB:");
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        System.out.print(AB[i][j] + " ");
        if (AB[i][j] != BA[i][j]){
            isEqual = false;
            break;
        }
    }
    System.out.println();
}

System.out.println(x:"Matrix BA:");
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) System.out.print(BA[i][j] + " ");
    System.out.println();
}

// System.out.println((isEqual) ? "AB = BA" : "AB ≠ BA");
if(isEqual){
    System.out.println(x:"AB = BA");
} else {
    System.out.println(x:"Ab ≠ BA");
}
sc.close();
}
```

```
PS E:\Java\Java_Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStorage\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assignment 1_c062fec8\bin' 'MatrixMultiplication'
Enter Matrix A (4x4):
2 4 6 8
1 3 5 7
3 5 7 9
2 3 5 6
Enter Matrix B (4x4):
1 2 3 4
5 6 7 8
9 8 7 6
5 4 3 2
Matrix AB:
116 108 100 92
96 88 80 72
136 128 120 112
92 86 80 74
Matrix BA:
21 37 57 73
53 97 149 193
59 113 173 227
27 53 81 107
AB ? BA
PS E:\Java\Java_Assignment 1> █
```

```
PS E:\Java\Java_Assignment 1> e:; cd 'e:\Java\Java_Assignment 1'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStorage\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assignment_1_c062fec8\bin' 'MatrixMultiplication'
Enter Matrix A (4x4):
23 34 45 56
56 67 78 89
12 45 78 90
67 54 32 87
Enter Matrix B (4x4):
54 67 89 12
89 56 12 34
14 52 36 47
42 35 46 57
Matrix AB:
7250 7745 6651 6739
13817 14675 12690 11689
9525 10530 8556 10470
12526 12222 11765 9103
Matrix BA:
6866 10978 14982 18041
7605 9154 10397 14006
6815 8118 8998 12741
7297 8921 10032 14566
AB ? BA
PS E:\Java\Java_Assignment 1> █
```

```
PS E:\Java\Java_Assignment 1> e:; cd 'e:\Java\Java_Assignment 1'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStorage\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assignment 1_c062fec8\bin' 'MatrixMultiplication'
Enter Matrix A (4x4):
15 26 37 44
13 11 44 25
21 32 43 54
65 76 87 98
Enter Matrix B (4x4):
96 45 65 34
2 3 4 5
56 4 76 8
34 7 56 89
Matrix AB:
5060 1209 6355 4852
4584 969 5633 3074
6324 1591 7785 6024
14596 4187 16629 12008
Matrix BA:
5600 7655 11285 12191
478 593 813 869
3008 4540 6212 7452
7562 9517 11717 13417
AB ? BA
PS E:\Java\Java_Assignment 1>
```

Sol

To solve the problem, follow the steps:

### Algorithm:

#### 1. Input the integer:

- Take an integer as input from the user.

#### 2. Check if the number is prime:

- A prime number is a number <sup>greater</sup> than 1 that has no divisions other than 1 and itself.
- Check if the number is divisible by any number between 2 and the square root of the number. If so, it is not prime.

#### 3. Binary Representation (if prime):

- If the number is prime, convert it to binary without using any built-in functions.
- Continuously divide the number by 2, keeping track of the remainder.
- Append each remainder (0,1) to a string, and then reverse the string to get the binary representation.

#### 4. Octal Representation (if NOT prime):

- If the number is not prime, convert it to octal without using any built-in function.

- Continuously divide the number by 8, keeping track of the remainder.
- Append each remainder (0-7) to a string, and then reverse the string to get the octal representation.

## CODE

```

import java.util.Scanner;
public class primeBinaryorOctal {
    public static boolean isPrime (int num) {
        if (num <= 1) return false;
        for (int i=2 ; i*i<= num ; i++)
            if (num % i == 0)
                return false;
        return true;
    }
}

Public static String toBinary (int num) {
    String binary = " ";
    while (num>0) {
        binary = (num % 2) + binary;
        num /= 2;
    }
    return binary;
}

```

```

Public static String toOctal (int num) {
    String octal = " ";
    while (num>0) {
        octal = (num % 8) + octal;
        num /= 8;
    }
}

```

6 Date \_\_\_\_\_  
Page \_\_\_\_\_

```
num / = 8;  
}
```

```
return octal;  
}
```

```
Public static void main (String [ ] args) {  
Scanner sc = new Scanner (System. in );  
System.out.println ("Enter the integer : ");  
int sum = sc.nextInt ();  
if (isPrime (num)) {  
    System.out.println ("The number is prime.  
    Binary : " + toBinary (num));  
}  
else  
{  
    System.out.println ("The number is not  
    prime. Octal : " + toOctal (num));  
}  
sc. close ();  
}
```

```
import java.util.Scanner;

public class PrimeBinaryOrOctal {

    // Check if a No. is prime
    public static boolean isPrime(int num) {
        if (num <= 1) return false;
        for (int i = 2; i * i <= num; i++) if (num % i == 0) return false;
        return true;
    }

    // Convert No. to binary
    public static String toBinary(int num) {
        String binary = "";
        while (num > 0) {
            binary = (num % 2) + binary;
            num /= 2;
        }
        return binary;
    }

    // Convert No. to octal
    public static String toOctal(int num) {
        String octal = "";
        while (num > 0) {
            octal = (num % 8) + octal;
            num /= 8;
        }
        return octal;
    }
}
```

```
Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter an integer: ");
    int num = sc.nextInt();

    if (isPrime(num)) {
        System.out.println("The number is prime. Binary: " + toBinary(num));
    } else {
        System.out.println("The number is not prime. Octal: " + toOctal(num));
    }

    sc.close();
}
```

```
PS E:\Java\Java_Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' 'ages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStor:2\redhat.java\jdt_ws\Java_Assignment 1_c062fec8\bin' 'PrimeBinaryOrOctal'  
Enter an integer: 5  
The number is prime. Binary: 101  
PS E:\Java\Java_Assignment 1> []
```

```
PS E:\Java\Java_Assignment 1>
PS E:\Java\Java_Assignment 1> e:; cd 'e:\Java\Java_Assignment 1'; & 'C:\Pro
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh choudhary\A
rage\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assignment 1_c
Enter an integer: 13
The number is prime. Binary: 1101
PS E:\Java\Java_Assignment 1> █
```

```
PS E:\Java\Java_Assignment 1> e:; cd 'e:\Java\Java_Assignment 1'; & 'C:\Program File  
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh choudhary\AppData\Ro  
rage\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assignment 1_c062fec8\|  
Enter an integer: 6  
The number is not prime. Octal: 6  
PS E:\Java\Java_Assignment 1> █
```

Q.8

To solve the problem, follow the steps:

Algorithm:

1. Input a Double:

- Take a double (floating-point) number as input from the user.

2. Ceiling operations:

- Use the ceiling function to compute the smallest integer greater than or equal to  $x$ , in java this can be done using `math.ceil(x)`.

3. 2's Complement:

- To calculate the 2's complement of number  $y$ .
- Convert the integer  $y$  to its binary representation.
- Invert the bits (1's Complements).
- Add 1 to the inverted bits to get the 2's complement.

4. Output

- Print the ceiling value  $y$ .
- Print the 2's Complement binary representation of  $y$ .

CODE

```

import java.util.Scanner;
public class Ceiling And Two's Complement {
    public static String two'sComplement(int num) {
        StringBuilder binary = new StringBuilder();
        for (int i = 31; i >= 0; i--) {
            binary.append((num & (1 << i)) != 0?
                "1": "0");
        }
        return binary.toString();
    }

    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter the decimal value : ");
        double x = sc.nextDouble();
        int y = (int) Math.ceil (x);
        System.out.println ("Ceiling value (y) = " + y);
        System.out.println ("2's Complement of y : "
            + two'sComplement (ny + 1));
        sc.close();
    }
}

```

```
import java.util.Scanner;

public class CeilingAndTwosComplement {

    // Function to get 2's complement of an integer
    public static String twosComplement(int num) {
        StringBuilder binary = new StringBuilder();
        for (int i = 31; i >= 0; i--) {
            binary.append((num & (1 << i)) != 0 ? "1" : "0");
        }
        return binary.toString();
    }

    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input double value
        System.out.print("Enter a decimal value: ");
        double x = sc.nextDouble();

        // Compute ceiling
        int y = (int) Math.ceil(x);

        // Print ceiling value
        System.out.println("Ceiling value (y) = " + y);

        // Print 2's complement of y
        System.out.println("2's Complement of y: " + twosComplement(~y + 1));

        sc.close();
    }
}
```

```
PS E:\Java\Java_Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+Showages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStorage\8f412\redhat.java\jdt_ws\Java_Assignment 1_c062fec8\bin' 'CeilingAndTwosComplement'  
Enter a decimal value: 45  
Ceiling value (y) = 45  
2's Complement of y: 1111111111111111111111111010011  
PS E:\Java\Java_Assignment 1> █
```

```
PS E:\Java\Java_Assignment 1> e:; cd 'e:\Java\Java_Assignment 1'  
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh  
rare\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assi  
nt'  
Enter a decimal value: 11  
Ceiling value (y) = 11  
2's Complement of y: 1111111111111111111111111111110101  
PS E:\Java\Java_Assignment 1> █
```

```
PS E:\Java\Java_Assignment 1> e;; cd 'e:\Java\Java_Assignment 1'; & 'C:\Program Files\Java\jdk-11.0.2\bin\java' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\mukesh choudhary\Appli  
rage\8f4111f2ec57be04b0890b092de97b72\redhat.java\jdt_ws\Java_Assignment 1_0001.jar' 'com.mukeshchoudhary.assignment1.CeilingValue'
```

```
Enter a decimal value: 15
```

```
Ceiling value (y) = 15
```

```
2's Complement of y: 1111111111111111111111110001
```

```
PS E:\Java\Java_Assignment 1> █
```

Q'4

Sol To solve this problem, follow the steps:

### Algorithm:-

#### 1. Design the flight class:

- Private attributes : Flight number, destination, departure, arrival time.
- Private method : check if the flight is upcoming based on current time.
- public methods:
  - update departure and arrival times.
  - display flight details.

#### 2. Design the airport class:

- Private attributes : Airport name and list of flights.
- Private method : find a flight by its flight number.
- Public methods:
  - Add a flight to the list.
  - Remove a flight by its flight number.
  - List upcoming flights (using the flights class private method).
  - List completed flights (using the flights class's private method).
  - display details of all flights.

### 3. Implement operations:

- Create instances of Airport and flight.
- Add, remove, update flight times, and display flight details.

### CODE

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

```
// class to represent a flight
```

```
class flight {
```

```
    private String flightNumber, destination,  
    String departureTime, String arrivalTime;
```

```
    public flight (String flightNumber, String destination,  
    String departureTime, String arrivalTime) {
```

```
        this.flightNumber = flightNumber;
```

```
        this.destination = destination;
```

```
        this.departureTime = departureTime;
```

```
        this.arrivalTime = arrivalTime;
```

```
    public void updateDepartureTime (String newTime) {  
        this.departureTime = newTime;
```

```
}
```

```
public String getFlightNumber () {
```

```
    return flightNumber;
```

```
}
```

```
public boolean isUpcoming () {  
    return departureTime - CompareTo (java.time.LocalDateTime.now ()) > 0;  
}
```

```
public void display () {  
    System.out.println ("Flight : " + flightNumber + "  
    To : " + destination + "; Departure : " + arrival : " +  
    arrivalTime);  
}
```

```
}
```

```
class Airport {  
    private ArrayList<Flight> flights = new ArrayList<> ();  
  
    public void addFlight (Flight flight) {  
        flights.add (flight);  
    }
```

```
    public void removeFlight (String flightNumber) {  
        flights.removeIf (f → f.getFlightNumber ().equals  
        (flightNumber));  
    }
```

```
    public void displayFlights (boolean upcoming) {  
        flights.stream ()  
            .filter (f → f.isUpcoming () == upcoming)  
            .filter.forEach (Flight::display);  
    }
```

```
    public void displayAllFlights () {  
        flights.forEach (Flight::display);  
    }
```

```
}
```

## Public class Airport Management

```
public static void main (String [ ] args) {  
    Airport airport = new Airport ();
```

```
airport.addFlight(new Flight(flightNumber: "Dg61",  
destination: "Los Angeles", departureTime: "03:00",  
arrival: "14:00"));
```

```
airport.addFlight(new Flight({flightNumber: "R945",  
destination: "RUSSIA", departureTime: "19:00",  
arrivalTime: "04:00"}));
```

```
System.out.println("All flights:");
airport.displayAllFlights();
```

```
System.out.println("In Upcoming Flights:");
airport.displayFlights(upcoming: true);
```

```
airport.removeFlight(FlightNumber: "A123");
```

```
System.out.println("In/After removing flight #s:")  
airport.displayAllFlights();
```

3

```
import java.util.ArrayList;

// Class to represent a Flight
class Flight {
    private String flightNumber, destination, departureTime, arrivalTime;

    // Constructor to initialize flight details
    public Flight(String flightNumber, String destination, String departureTime, String arrivalTime) {
        this.flightNumber = flightNumber;
        this.destination = destination;
        this.departureTime = departureTime;
        this.arrivalTime = arrivalTime;
    }

    // Method to update the departure time of the flight
    public void updateDepartureTime(String newTime) {
        this.departureTime = newTime;
    }

    public String getFlightNumber() {
        return flightNumber;
    }

    // Method to check if the flight is upcoming based on current time
    public boolean isUpcoming() {
        return departureTime.compareTo(java.time.LocalDateTime.now().toString()) > 0;
    }

    public void display() {
        System.out.println("Flight: " + flightNumber + ", To: " + destination +
                           ", Departure: " + departureTime + ", Arrival: " + arrivalTime);
    }
}

class Airport {
    private ArrayList<Flight> flights = new ArrayList<>();

    public void addFlight(Flight flight) {
        flights.add(flight);
    }
}
```

```
class Airport {  
    private ArrayList<Flight> flights = new ArrayList<>();  
  
    public void addFlight(Flight flight) {  
        flights.add(flight);  
    }  
    public void removeFlight(String flightNumber) {  
        flights.removeIf(f -> f.getFlightNumber().equals(flightNumber));  
    }  
  
    public void displayFlights(boolean upcoming) {  
        flights.stream()  
            .filter(f -> f.isUpcoming() == upcoming)  
            .forEach(Flight::display);  
    }  
  
    public void displayAllFlights() {  
        flights.forEach(Flight::display);  
    }  
}  
  
public class AirportManagement {  
    Run | Debug  
    public static void main(String[] args) {  
        Airport airport = new Airport(); // Create an Airport instance  
  
        airport.addFlight(new Flight(flightNumber:"D961", destination:"LOS ANGELES", departureTime:"03:00", arrivalTime:"14:00"));  
        airport.addFlight(new Flight(flightNumber:"B246", destination:"RUSSIA", departureTime:"19:00", arrivalTime:"04:00"));  
  
        System.out.println("All Flights:");  
        airport.displayAllFlights();  
  
        System.out.println("\nUpcoming Flights:");  
        airport.displayFlights(upcoming:true);  
  
        airport.removeFlight(flightNumber:"A123");  
        System.out.println("\nAfter removing flight A123:");  
        airport.displayAllFlights(); // Display all flights after removal  
    }  
}
```

```
PS E:\Java\Java_Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDeta:  
ges' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStorage\8f4111f2ec57be  
2\redhat.java\jdt_ws\Java_Assignment 1_c062fec8\bin' 'AirportManagement'
```

All Flights:

```
Flight: D961, To: LOS ANGELES, Departure: 03:00, Arrival: 14:00
```

```
Flight: B246, To: RUSSIA, Departure: 19:00, Arrival: 04:00
```

Upcoming Flights:

```
Flight: B246, To: RUSSIA, Departure: 19:00, Arrival: 04:00
```

After removing flight A123:

```
Flight: D961, To: LOS ANGELES, Departure: 03:00, Arrival: 14:00
```

```
Flight: B246, To: RUSSIA, Departure: 19:00, Arrival: 04:00
```

```
PS E:\Java\Java_Assignment 1>
```

Q.5

Sol

## Algorithms

### 1. Define the shape Interface

- Create an interface named Shape.
- Declare a method area() that returns a double.

### 2. Create the circle class

- Implement the shape interface.
- Define a private attribute for radius and a constant for pi.
- Create a constructor to initialize the radius.
- Implement the area() method to calculate and return the area of the circle using the formula.

### 3. Create the rectangle Class

- Implement the shape interface.
- Define private attribute for length and width.
- Create a constructor to initialize length and width.
- Implement the area() method to calculate and return the area of the rectangle.

### 4. Create the triangle Class

- Implement the Shape interface
- Define private attributes for base and height.
- Create a constructor to initialize base and height.
- Implement the area() method to calculate and return the area of the triangle using the formula.

## 5. Create the ShapeAreaCalculator class

- Define a method calculate-area (Shape shape) that
  - Takes a Shape object as a parameter.
  - calls the area() method on the passed shape object.
  - Return the calculated area.

## 6. Main program (GeometricShapeAreaCalculator)

- Initialize Scanner for user input.
- Loop to display a menu:
  - Prompt user for shape choice.
  - if choice is 1-3, gather dimensions and create corresponding shape object.
  - if choice is 4, exit the program.
- Close the scanner.

CODE :-

```
import java.util.Scanner;  
interface Shape {  
    double area();  
}  
  
class Circle implements Shape {  
    private double radius;  
    private final double pi = math.PI;  
  
    public Circle (double radius) {  
        this.radius = radius;  
    }
```

@override

public double area () {

return pi \* radius \* radius;

}

}

class Rectangle implements Shape {

private double length;

private double width;

public Rectangle (double length, double width) {

this.length = length;

this.width = width;

}

@Override

public double area () {

return length \* width;

}

class Triangle implements Shape {

private double base;

private double height;

public Triangle (double base, double height) {

this.base = base;

this.height = height;

}

@Override

public double area () {

return 0.5 \* base \* height;

}

```
class ShapeAreaCalculator {  
    public double calculateArea(Shape shape) {  
        return shape.area();  
    }  
}
```

```
public class GeometricShapeAreaCalculator {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        ShapeAreaCalculator calculator = new ShapeAreaCalculator();  
  
        while (true) {  
            System.out.println("Select a shape to calculate the area:");  
            System.out.println("1. Circle");  
            System.out.println("2. Rectangle");  
            System.out.println("3. Triangle");  
            System.out.println("4. Exit");  
            System.out.println("Enter your choice (1-4):");  
            int choice = scanner.nextInt();  
  
            if (choice == 4) {  
                System.out.println("Exiting the program.  
                Goodbye!");  
                break;  
            }  
            Shape shape = null;  
  
            switch (choice) {  
                case 1:  
                    System.out.print("Enter the radius of the  
                    circle: ");
```

```
        double radius = scanner.nextDouble();
        Shape = new Circle(radius);
        break;
```

Case 2:

```
System.out.println("Enter the length of the rectangle:");
double length = scanner.nextDouble();
System.out.print("Enter the width of the rectangle:");
double width = scanner.nextDouble();
Shape = new Rectangle(length, width);
break;
```

Case 3:

```
System.out.print("Enter the base of the triangle:");
double base = scanner.nextDouble();
System.out.print("Enter the height of the triangle:");
double height = scanner.nextDouble();
Shape = new Triangle(base, height);
break;
```

default :

```
System.out.println("Invalid choice! please select again");
Continue;
```

```
} }
```

```
double area = calculator.calculateArea(Shape);
```

```
System.out.println("The area is: " + area);
```

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

```
}
```

```
scanner.close();
```

```
}
```

```
}
```

```
import java.util.Scanner;

// Step 1: Create the Shape interface
interface Shape {
    double area();
}

// Step 2: Implement Circle class
class Circle implements Shape {
    private double radius;
    private final double pi = Math.PI;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double area() {
        return pi * radius * radius;
    }
}

// Step 3: Implement Rectangle class
class Rectangle implements Shape {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    @Override
    public double area() {
        return length * width;
    }
}
```

```
// Step 4: Implement Triangle class
class Triangle implements Shape {
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    public double area() {
        return 0.5 * base * height;
    }
}

// Step 5: Create ShapeAreaCalculator class
class ShapeAreaCalculator {
    public double calculate_area(Shape shape) {
        return shape.area();
    }
}

// Step 6: Implement the main class for user interaction
public class GeometricShapeAreaCalculator {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ShapeAreaCalculator calculator = new ShapeAreaCalculator();

        while (true) {
            System.out.println("Select a shape to calculate the area:");
            System.out.println("1. Circle");
            System.out.println("2. Rectangle");
            System.out.println("3. Triangle");
            System.out.println("4. Exit");
            System.out.print("Enter your choice (1-4): ");
            int choice = scanner.nextInt();
        }
    }
}
```

```
if (choice == 4) {
    System.out.println("Exiting the program. Goodbye!");
    break;
}

Shape shape = null; // Variable to hold the shape object

switch (choice) {
    case 1: // Circle
        System.out.print("Enter the radius of the circle: ");
        double radius = scanner.nextDouble();
        shape = new Circle(radius);
        break;

    case 2: // Rectangle
        System.out.print("Enter the length of the rectangle: ");
        double length = scanner.nextDouble();
        System.out.print("Enter the width of the rectangle: ");
        double width = scanner.nextDouble();
        shape = new Rectangle(length, width);
        break;

    case 3: // Triangle
        System.out.print("Enter the base of the triangle: ");
        double base = scanner.nextDouble();
        System.out.print("Enter the height of the triangle: ");
        double height = scanner.nextDouble();
        shape = new Triangle(base, height);
        break;

    default:
        System.out.println("Invalid choice! Please select again.");
        continue;
}
```

```
        }  
    }  
  
    default:  
        System.out.println(x:"Invalid choice! Please select again.");  
        continue;  
    }  
  
    // Calculate and display the area of the chosen shape  
    double area = calculator.calculate_area(shape);  
    System.out.println("The area is: " + area);  
    System.out.println(); // Print a new line for better readability  
}  
  
scanner.close(); // Close the scanner  
}
```

```
PS E:\Java\Java_Assignment_1> & 'C:\Program Files\Java\jdk-21\bin  
ages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\w  
2\redhat.java\jdt_ws\Java_Assignment_1_c062fec8\bin' 'GeometricSha  
Select a shape to calculate the area:  
1. Circle  
2. Rectangle  
3. Triangle  
4. Exit  
Enter your choice (1-4): 2  
Enter the length of the rectangle: 6  
Enter the width of the rectangle: 8  
The area is: 48.0
```

```
PS E:\Java\Java_Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\2\redhat.java\jdt_ws\Java_Assignment_1_c062fec8\bin' 'GeometricShapes'
Select a shape to calculate the area:
1. Circle
2. Rectangle
3. Triangle
4. Exit
Enter your choice (1-4): 3
Enter the base of the triangle: 7
Enter the height of the triangle: 12
The area is: 42.0
```

```
PS E:\Java\Java_Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:  
ages' '-cp' 'C:\Users\mukesh choudhary\AppData\Roaming\Code\User\workspaceStorage  
2\redhat.java\jdt_ws\Java_Assignment_1_c062fec8\bin' 'GeometricShapeAreaCalculato  
Select a shape to calculate the area:
```

- 1. Circle
- 2. Rectangle
- 3. Triangle
- 4. Exit

```
Enter your choice (1-4): 1
```

```
Enter the radius of the circle: 12
```

```
The area is: 452.3893421169302
```

```
Select a shape to calculate the area:
```

- 1. Circle
- 2. Rectangle
- 3. Triangle
- 4. Exit

```
Enter your choice (1-4): 4
```

```
Exiting the program. Goodbye!
```

```
PS E:\Java\Java_Assignment 1> █
```