**Final**

**Practice questions for Practicals - 1**

1. Java Program to Calculate Average Using Arrays

Code :

import java.util.Scanner;

public class averagecalculator {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n;

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

        n = input.nextInt();

        int[] numbers = new int[n];

        System.out.println("Enter the elements: ");

        for (int i = 0; i < n; i++) {

            numbers[i] = input.nextInt();

        }

        double sum = 0;

        for (int i = 0; i < n; i++) {

            sum += numbers[i];

        }

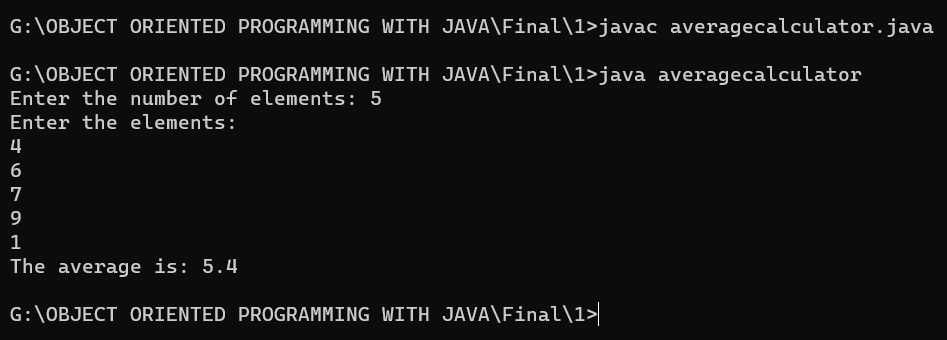
        double average = sum / n;

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

    }

}

Output :



1. Java Program to Find Largest Element of an Array

Code :

import java.util.Scanner;

public class averagecalculator {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n;

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

        n = input.nextInt();

        int[] numbers = new int[n];

        System.out.println("Enter the elements: ");

        for (int i = 0; i < n; i++) {

            numbers[i] = input.nextInt();

        }

        double sum = 0;

        for (int i = 0; i < n; i++) {

            sum += numbers[i];

        }

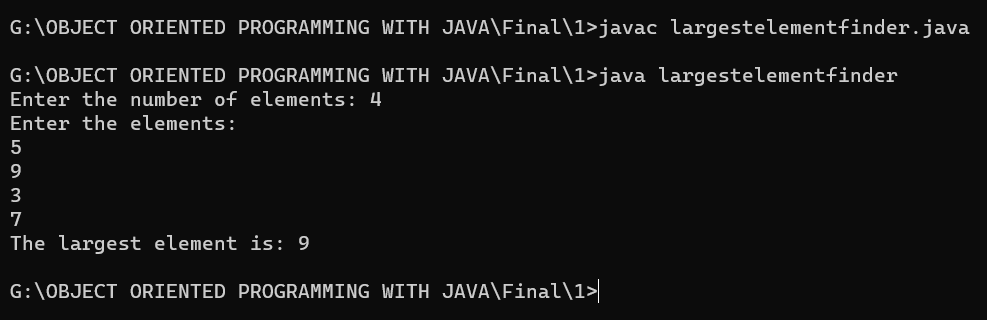
        double average = sum / n;

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

    }

}

Output :



1. Java Program to Add Two Matrix Using Multi-dimensional Arrays

Code :

import java.util.Scanner;

public class matrixaddition {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int rows, columns;

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

        rows = input.nextInt();

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

        columns = input.nextInt();

        int[][] matrix1 = new int[rows][columns];

        int[][] matrix2 = new int[rows][columns];

        int[][] resultMatrix = new int[rows][columns];

        System.out.println("Enter the elements of the first matrix:");

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

            for (int j = 0; j < columns; j++) {

                matrix1[i][j] = input.nextInt();

            }

        }

        System.out.println("Enter the elements of the second matrix:");

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

            for (int j = 0; j < columns; j++) {

                matrix2[i][j] = input.nextInt();

            }

        }

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

            for (int j = 0; j < columns; j++) {

                resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

            }

        }

        System.out.println("The result of the addition is:");

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

            for (int j = 0; j < columns; j++) {

                System.out.print(resultMatrix[i][j] + " ");

            }

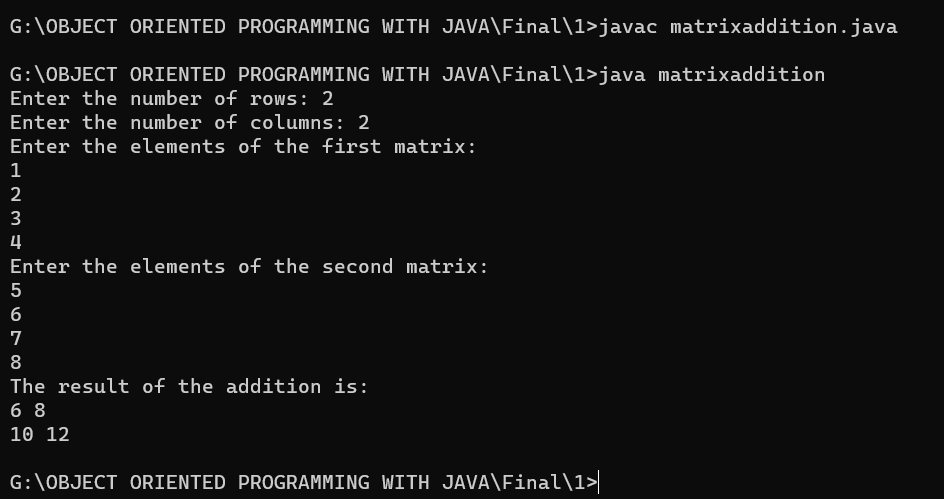
            System.out.println();

        }

    }

}

Output :



1. Java Program to Find the Sum of Natural Numbers using Recursion

Code :

import java.util.Scanner;

public class NaturalNumberSum {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n;

        System.out.print("Enter a positive integer: ");

        n = input.nextInt();

        int sum = findSum(n);

        System.out.println("The sum of the first " + n + " natural numbers is: " + sum);

    }

    public static int findSum(int n) {

        if (n == 1) {

            return 1;

        } else {

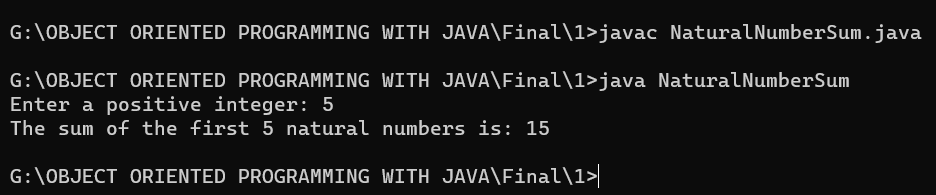
            return n + findSum(n-1);

        }

    }

}

Output :



1. Java Program to Reverse a Sentence Using Recursion

Code :

import java.util.Scanner;

public class SentenceReverser {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        String sentence;

        System.out.print("Enter a sentence: ");

        sentence = input.nextLine();

        String reversed = reverseSentence(sentence);

        System.out.println("The reversed sentence is: " + reversed);

    }

    public static String reverseSentence(String sentence) {

        if (sentence.isEmpty()) {

            return sentence;

        } else {

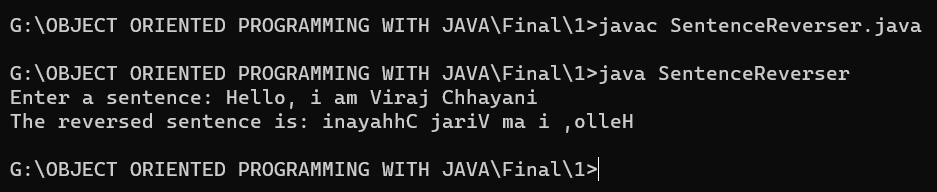
            return reverseSentence(sentence.substring(1)) + sentence.charAt(0);

        }

    }

}

Output :



1. Write a java program to find the Fibonacci series using recursive and non recursive functions.

Code :

import java.util.Scanner;

public class Fibonacci {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int n;

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

        n = input.nextInt();

        System.out.println("Using non-recursive function:");

        for (int i = 0; i < n; i++) {

            System.out.print(fibonacciNonRecursive(i) + " ");

        }

        System.out.println();

        System.out.println("Using recursive function:");

        for (int i = 0; i < n; i++) {

            System.out.print(fibonacciRecursive(i) + " ");

        }

        System.out.println();

    }

    public static int fibonacciNonRecursive(int n) {

        if (n == 0 || n == 1) {

            return n;

        } else {

            int fib0 = 0;

            int fib1 = 1;

            int fib = 0;

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

                fib = fib0 + fib1;

                fib0 = fib1;

                fib1 = fib;

            }

            return fib;

        }

    }

    public static int fibonacciRecursive(int n) {

        if (n == 0 || n == 1) {

            return n;

        } else {

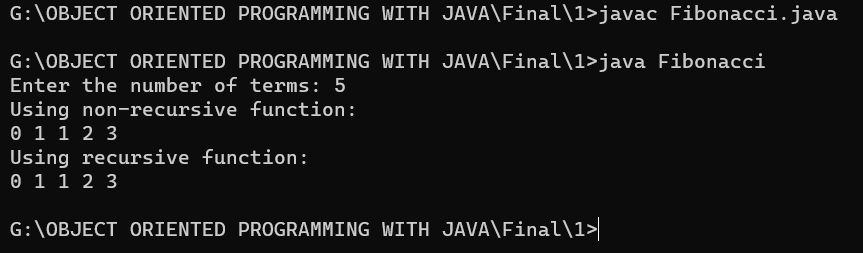
            return fibonacciRecursive(n-1) + fibonacciRecursive(n-2);

        }

    }

}

Output :



1. Java Program to Calculate Difference Between Two Time Periods

Code :

import java.util.Scanner;

public class TimeDifference {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int hours1, minutes1, seconds1, hours2, minutes2, seconds2;

        System.out.print("Enter time period 1 (hours minutes seconds): ");

        hours1 = input.nextInt();

        minutes1 = input.nextInt();

        seconds1 = input.nextInt();

        System.out.print("Enter time period 2 (hours minutes seconds): ");

        hours2 = input.nextInt();

        minutes2 = input.nextInt();

        seconds2 = input.nextInt();

        int diffHours = hours2 - hours1;

        int diffMinutes = minutes2 - minutes1;

        int diffSeconds = seconds2 - seconds1;

        if (diffSeconds < 0) {

            diffMinutes--;

            diffSeconds += 60;

        }

        if (diffMinutes < 0) {

            diffHours--;

            diffMinutes += 60;

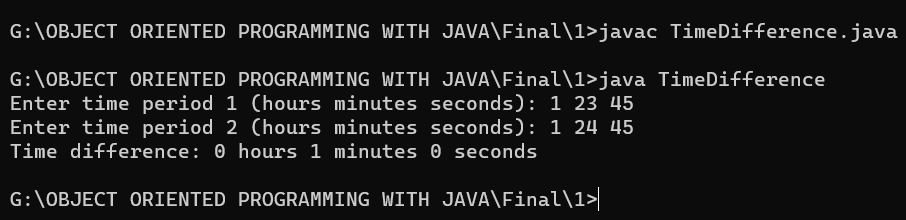
        }

        System.out.println("Time difference: " + diffHours + " hours " + diffMinutes + " minutes " + diffSeconds + " seconds");

    }

}

Output :



1. Java Program to Check if An Array Contains a Given Value

Code :

import java.util.Scanner;

public class ArrayContainsValue {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter the size of the array: ");

        int size = input.nextInt();

        int[] arr = new int[size];

        System.out.println("Enter the elements of the array: ");

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

            arr[i] = input.nextInt();

        }

        System.out.print("Enter the value to search for: ");

        int value = input.nextInt();

        boolean found = false;

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

            if (arr[i] == value) {

                found = true;

                break;

            }

        }

        if (found) {

            System.out.println("The array contains the value " + value);

        } else {

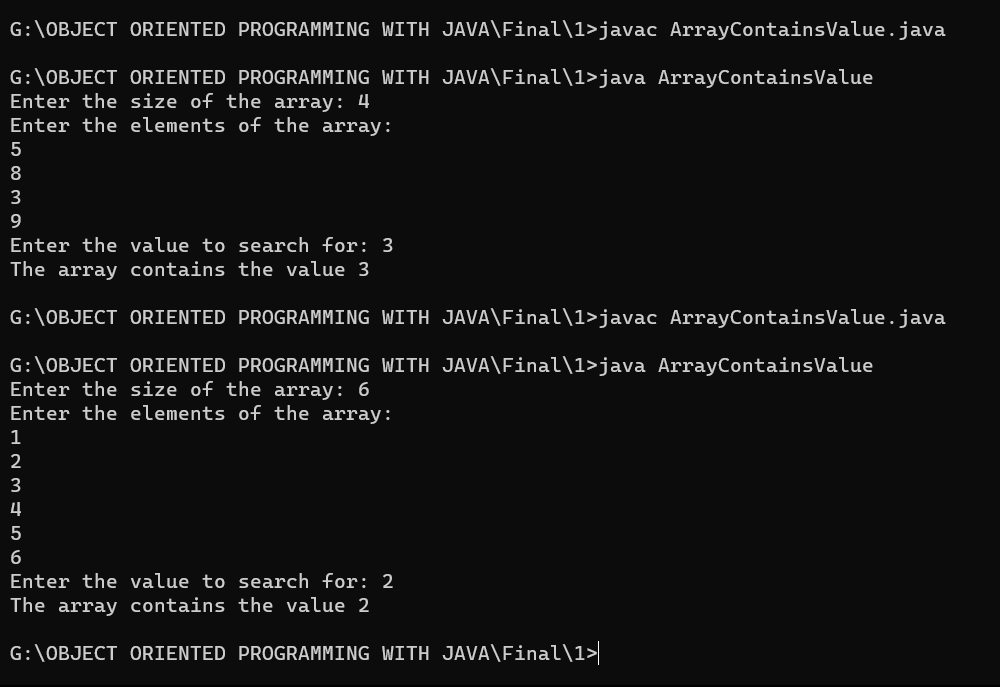
            System.out.println("The array does not contain the value " + value);

        }

    }

}

Output :



1. Java Program to Round a Number to n Decimal Places

Code :

import java.util.Scanner;

public class RoundNumber {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter a number: ");

        double number = input.nextDouble();

        System.out.print("Enter the number of decimal places to round to: ");

        int decimalPlaces = input.nextInt();

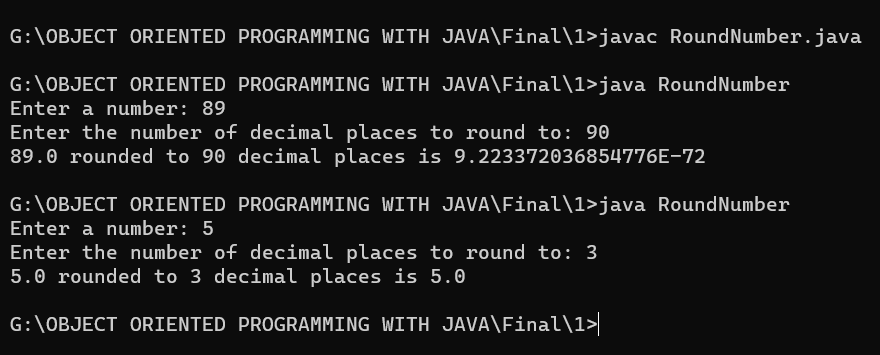
        double roundedNumber = Math.round(number \* Math.pow(10, decimalPlaces)) / Math.pow(10, decimalPlaces);

        System.out.println(number + " rounded to " + decimalPlaces + " decimal places is " + roundedNumber);

    }

}

Output :

****

**Practice questions for Practicals – 2**

1. Write a java program to display the employee details such as employee id, employee name, salary,age using Scanner class.

Code :

import java.util.Scanner;

public class EmployeeDetails {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter employee id: ");

        int empId = input.nextInt();

        input.nextLine(); *// consume the newline character*

        System.out.print("Enter employee name: ");

        String empName = input.nextLine();

        System.out.print("Enter employee salary: ");

        double empSalary = input.nextDouble();

        System.out.print("Enter employee age: ");

        int empAge = input.nextInt();

        System.out.println("Employee details:");

        System.out.println("ID: " + empId);

        System.out.println("Name: " + empName);

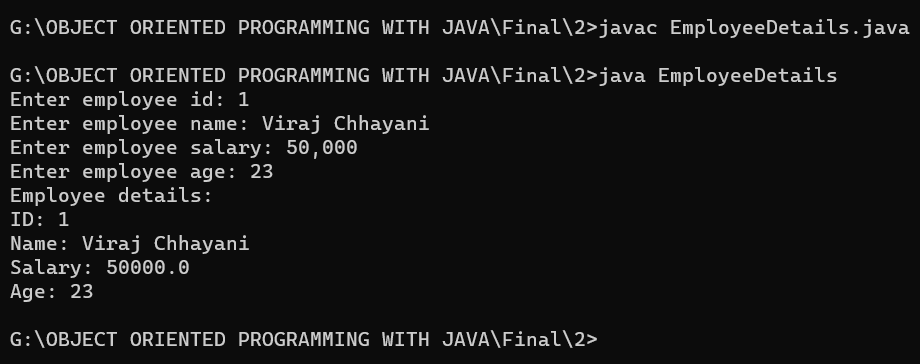
        System.out.println("Salary: " + empSalary);

        System.out.println("Age: " + empAge);

    }

}

Output :



1. Create an abstract class named shape that contains two integers and an empty method named printArea .Provide three classes named Rectangle ,Triangle and Circle subclass that each one of the classes extends the Class Shape .Each one of the classes contains only the method printArea() that prints the area of Shape.

Code :

abstract class Shape {

    protected int width;

    protected int height;

    public Shape(int width, int height) {

*this*.width = width;

*this*.height = height;

    }

    public abstract void printArea();

}

class Rectangle extends Shape {

    public Rectangle(int width, int height) {

*super*(width, height);

    }

    public void printArea() {

        int area = width \* height;

        System.out.println("Area of rectangle: " + area);

    }

}

class Triangle extends Shape {

    public Triangle(int width, int height) {

*super*(width, height);

    }

    public void printArea() {

        int area = width \* height / 2;

        System.out.println("Area of triangle: " + area);

    }

}

class Circle extends Shape {

    public Circle(int width, int height) {

*super*(width, height);

    }

    public void printArea() {

        double radius = width / 2.0;

        double area = Math.PI \* radius \* radius;

        System.out.println("Area of circle: " + area);

    }

}

public class AllShape {

    public static void main(String[] args) {

        Rectangle r = new Rectangle(5, 10);

        r.printArea();

        Triangle t = new Triangle(5, 10);

        t.printArea();

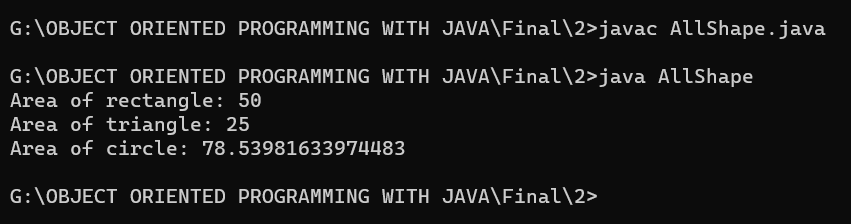
        Circle c = new Circle(10, 0);

        c.printArea();

    }

}

Output :



1. Design a class to represent a bank account. Which include contains account number, name of the depositor, type of the account, balance amount in the account. Define Methods, to assign initial values, to Deposit an amount, to Withdraw amount after checking balance, to display name and balance.(Hint: method overloading).

Code :

import java.util.Scanner;

public class BankAccount {

    private String accountNumber;

    private String name;

    private String accountType;

    private double balance;

    public BankAccount(String accountNumber, String name, String accountType, double balance) {

*this*.accountNumber = accountNumber;

*this*.name = name;

*this*.accountType = accountType;

*this*.balance = balance;

    }

    public BankAccount(String accountNumber, String name, String accountType) {

*this*(accountNumber, name, accountType, 0);

    }

    public void deposit(double amount) {

        balance += amount;

        System.out.println(amount + " deposited into account number " + accountNumber + ". New balance: " + balance);

    }

    public void withdraw(double amount) {

        if (balance < amount) {

            System.out.println("Insufficient funds to withdraw " + amount + " from account number " + accountNumber);

        } else {

            balance -= amount;

            System.out.println(amount + " withdrawn from account number " + accountNumber + ". New balance: " + balance);

        }

    }

    public void displayBalance() {

        System.out.println("Account number: " + accountNumber + ", balance: " + balance);

    }

    public void displayInfo() {

        System.out.println("Account number: " + accountNumber);

        System.out.println("Name: " + name);

        System.out.println("Account type: " + accountType);

        System.out.println("Balance: " + balance);

    }

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter account number: ");

        String accountNumber = input.nextLine();

        System.out.print("Enter name of depositor: ");

        String name = input.nextLine();

        System.out.print("Enter type of account: ");

        String accountType = input.nextLine();

        System.out.print("Enter initial balance: ");

        double balance = input.nextDouble();

        BankAccount account = new BankAccount(accountNumber, name, accountType, balance);

        System.out.print("Enter amount to deposit: ");

        double depositAmount = input.nextDouble();

        account.deposit(depositAmount);

        System.out.print("Enter amount to withdraw: ");

        double withdrawAmount = input.nextDouble();

        account.withdraw(withdrawAmount);

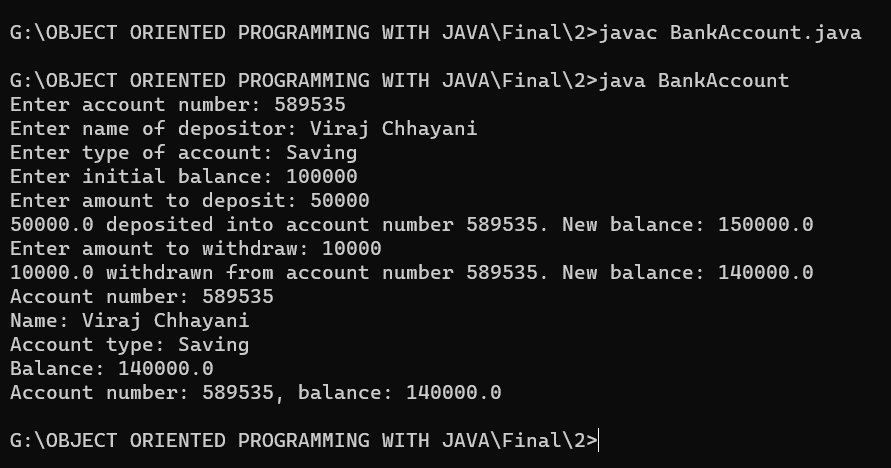
        account.displayInfo();

        account.displayBalance();

    }

}

Output :



1. Create class Vehicle having run ( ) method. Create class Bike that extends Vehicle class and by defining run () method demonstrate method overriding.

Code :

*// Vehicle class*

class Vehicle {

    public void run() {

        System.out.println("Vehicle is running");

    }

}

*// Bike class that extends Vehicle class*

class Bike extends Vehicle {

    @Override

    public void run() {

        System.out.println("Bike is running");

    }

}

*// Main class to test the method overriding*

public class AllVehicle {

    public static void main(String[] args) {

*// create a Vehicle object and call the run method*

        Vehicle vehicle = new Vehicle();

        vehicle.run();

*// create a Bike object and call the run method*

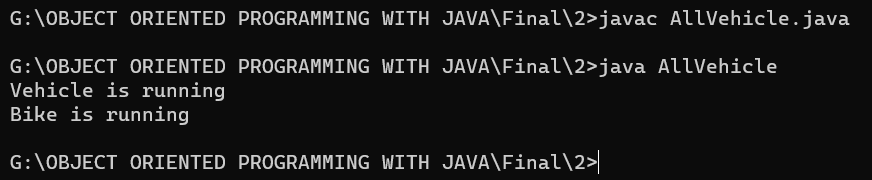
        Bike bike = new Bike();

        bike.run();

    }

}

Output :



1. Creates classes Car, Maruti and Maruti800. We have done a setup – class Maruti extends Car and class Maruti800 extends Maruti. With the help of this Multilevel hierarchy setup our Maruti800 class is able to use the methods of both the classes (Car and Maruti).

Code :

public class Maruti800 extends Maruti {

    public void fuelType() {

        System.out.println("Petrol");

    }

    public static void main(String[] args) {

        Maruti800 m800 = new Maruti800();

        m800.start();

        m800.accelerate();

        m800.brake();

        m800.stop();

        m800.fuelType();

    }

}

public class Car {

    public void start() {

        System.out.println("Car started");

    }

    public void stop() {

        System.out.println("Car stopped");

    }

}

public class Maruti extends Car {

    public void accelerate() {

        System.out.println("Maruti accelerating");

    }

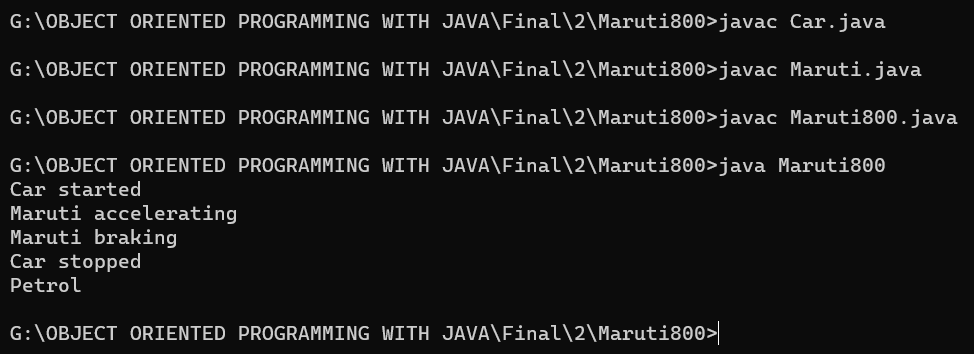
    public void brake() {

        System.out.println("Maruti braking");

    }

}

Output :



1. Do as Directed.
   * In a single inheritance.java file declare one variable as final. Try to change value of the final variable and Take Screenshot of error or output.
   * Implement JAVA program that show  Multilevel inheritance with student information.
   * Create a Class a,b,c,d where class b,c,d extends class a.(Hierarchical inheritance)

Code :

public class FinalVariableExample {

    public static void main(String[] args) {

        final int num = 10;

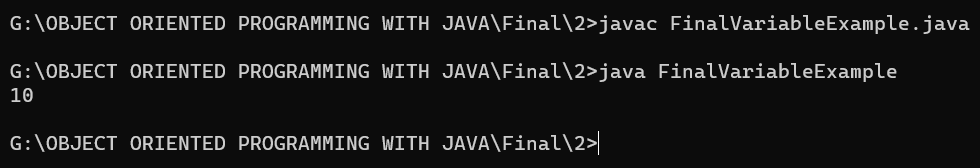
*//num = 20;  // This will give a compilation error since num is a final variable and its value cannot be changed*

        System.out.println(num);

    }

}

Output :



Code :

class Student {

    private String name;

    private int rollNo;

    public void setName(String name) {

*this*.name = name;

    }

    public String getName() {

        return name;

    }

    public void setRollNo(int rollNo) {

*this*.rollNo = rollNo;

    }

    public int getRollNo() {

        return rollNo;

    }

}

class Exam extends Student {

    private int marks;

    public void setMarks(int marks) {

*this*.marks = marks;

    }

    public int getMarks() {

        return marks;

    }

}

class Result extends Exam {

    public void display() {

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

        System.out.println("Roll No: " + getRollNo());

        System.out.println("Marks: " + getMarks());

    }

}

public class MultilevelInheritanceExample {

    public static void main(String[] args) {

        Result resultObj = new Result();

        resultObj.setName("Vxyz");

        resultObj.setRollNo(101);

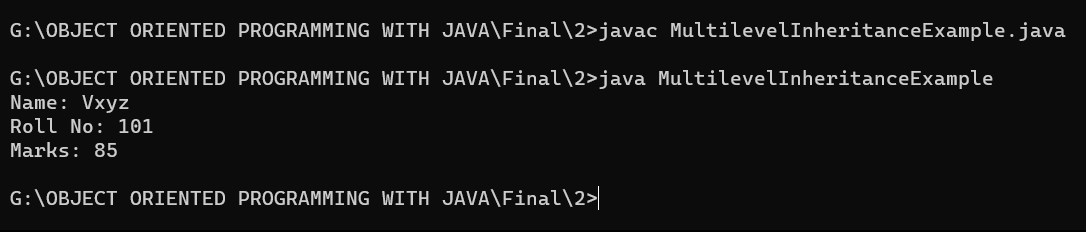
        resultObj.setMarks(85);

        resultObj.display();

    }

}

Output :



Code :

*// Class A*

class A {

    public void display() {

        System.out.println("This is class A");

    }

}

*// Class B extends A*

class B extends A {

    public void display() {

        System.out.println("This is class B");

    }

}

*// Class C extends A*

class C extends A {

    public void display() {

        System.out.println("This is class C");

    }

}

*// Class D extends A*

class D extends A {

    public void display() {

        System.out.println("This is class D");

    }

}

*// Main class*

class HierarchyExample {

    public static void main(String[] args) {

        A a = new A();

        B b = new B();

        C c = new C();

        D d = new D();

*// Displaying class A*

        a.display();

*// Displaying class B*

        b.display();

*// Displaying class C*

        c.display();

*// Displaying class D*

        d.display();

    }

}

Output :

