##### **Write a program to find total, average of given two numbers by using function with command-line arguments, static data members.**

public class **Calculator**

{

// Static data members

static int num1, num2,total;

static double average;

// Function to add the numbers and return the total

public static int **addNumbers()**

{

return num1 + num2;

}

// Function to calculate the average and return the average

public static double **calculateAverage()**

{

return (num1 + num2) / 2.0;

}

// Function to display results

public static void **displayResults()**

{

System.out.println("Total: " + total);

System.out.println("Average: " + average);

}

public static void **main**(String args[])

{

if (args.length != 2)

{

System.out.println("please pass num1 and num2 as commandline arguments");

return;

}

num1 = Integer.parseInt(args[0]);

num2 = Integer.parseInt(args[1]);

total = addNumbers();

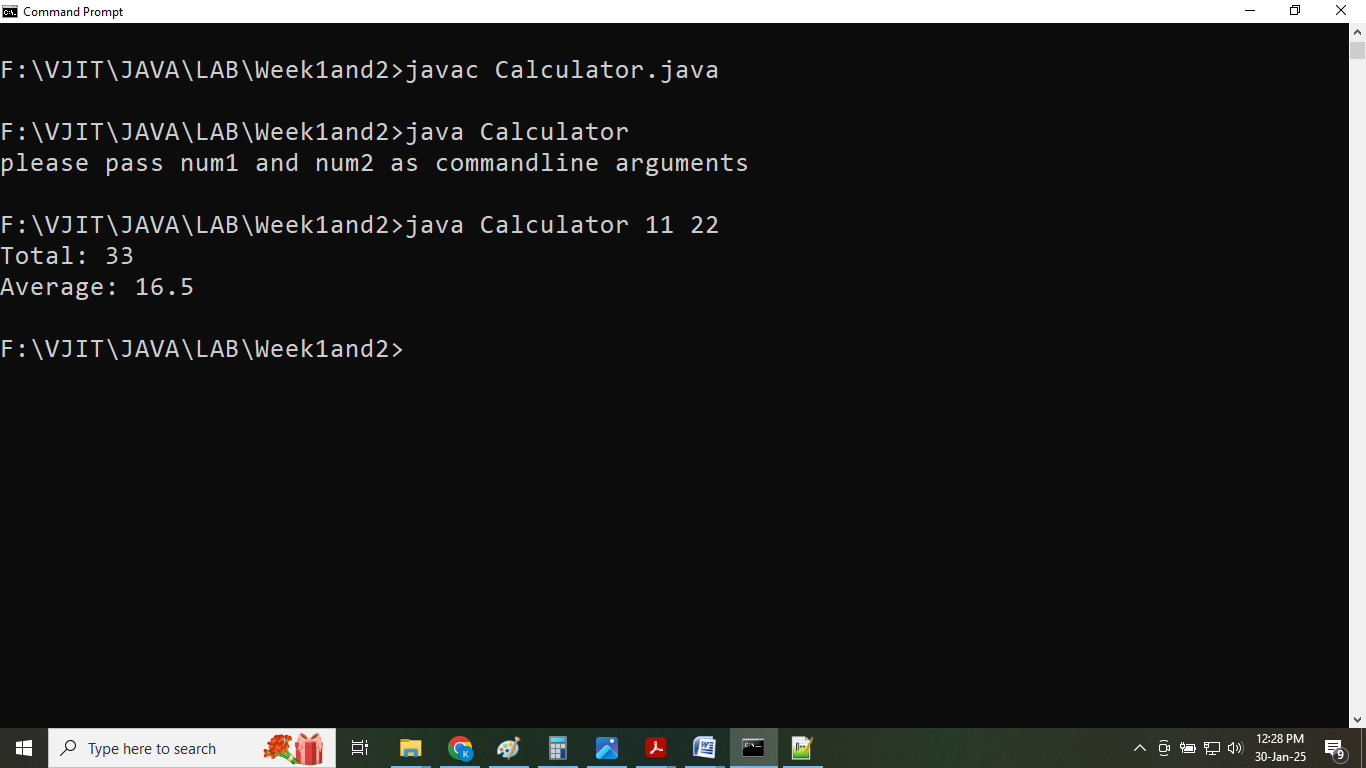
average = calculateAverage();

displayResults();

}

}

**Output:**



##### **Write a program to illustrate class and objects.**

import java.util.Scanner;

class **ObjectCounter**

{

// Static variable to keep track of the number of objects

private static int count = 0;

public **ObjectCounter()**

{

// Constructor increments the object count when an object is created

count++;

System.out.println("Object is created:"+count);

}

public static int **getObjectCount()**

{ // Static method to get the current count of objects & return count

return count;

}

}

public class **ObjectCountTest**

{

public static void **main**(String[] args)

{

Scanner sc = new Scanner(System.in);

// Prompt user for the number of objects to create

System.out.print("Enter the number of objects to create: ");

int numObjects = sc.nextInt();

// Loop to create the specified number of objects

for (int i = 0; i < numObjects; i++)

{

new ObjectCounter(); // Creating an object will increment the count

}

// Display the total count of objects created

System.out.println("Total number of objects created: " + ObjectCounter.getObjectCount());

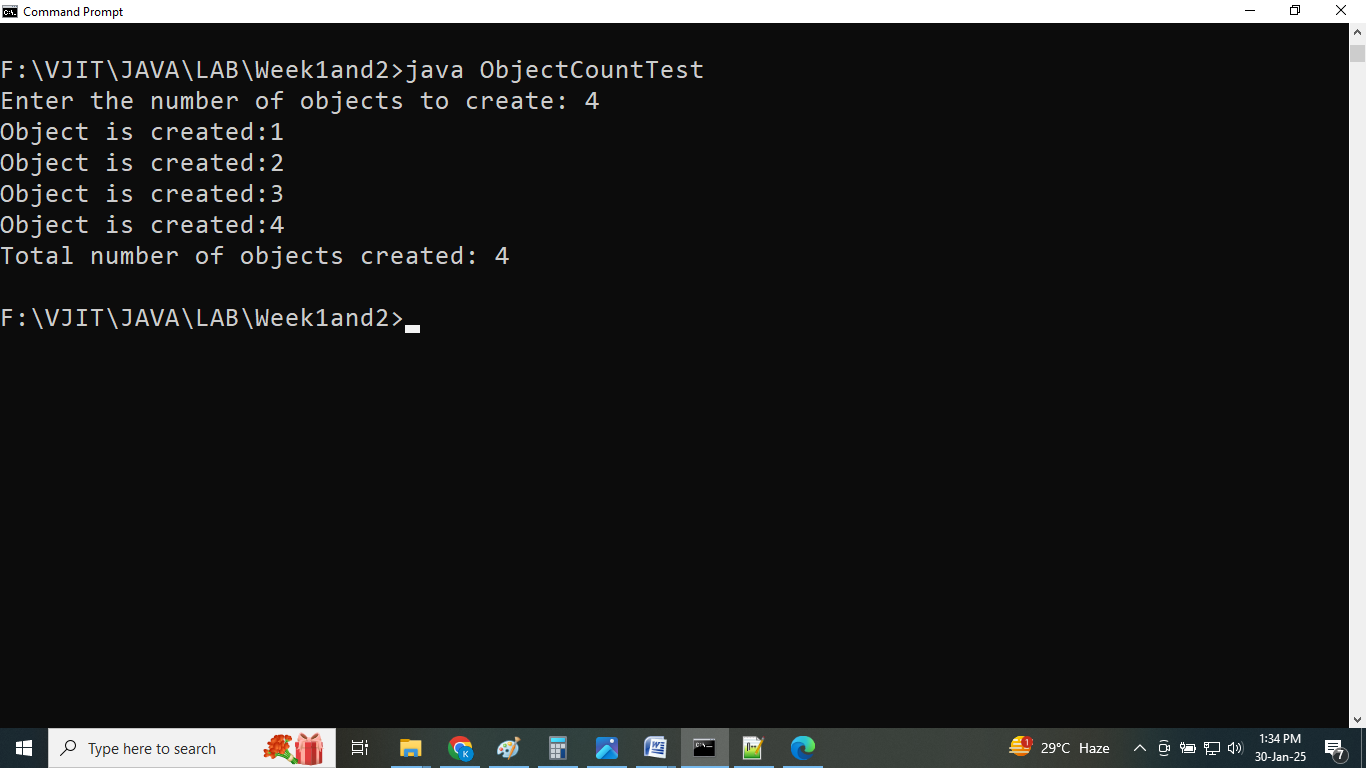
// Close the scanner

sc.close();

}

}

**Output:**



##### **Write a program to illustrate constructor overloading.**

import java.util.Scanner;

public class **Book**

{

// Attributes of the Book class

private String title;

private String author;

private int pageCount;

// Default constructor

public **Book()**

{

this.title = "Unknown Title";

this.author = "Unknown Author";

this.pageCount = 666;

}

// Constructor with title and author parameters

public **Book(String title, String author)**

{

this.title = title;

this.author = author;

//this.pageCount = 0; // Default page count

}

// Constructor with title, author, and pageCount parameters

public **Book(String title, String author, int pageCount)**

{

this.title = title;

this.author = author;

this.pageCount = pageCount;

}

// Method to display book details

public void **displayBookDetails()**

{

System.out.println("Title: " + title);

System.out.println("Author: " + author);

System.out.println("Page Count: " + pageCount);

}

public static void **main**(String[] args)

{

Scanner scanner = new Scanner(System.in);

// Taking user inputs for Book 1

System.out.println("Details for Book 1:");

System.out.print("Title: ");

String title1 = scanner.nextLine();

System.out.print("Author: ");

String author1 = scanner.nextLine();

// Creating Book 1 using the constructor with title and author parameters

Book book1 = new Book();

// Taking user inputs for Book 2

System.out.println("Details for Book 2:");

System.out.print("Title: ");

String title2 = scanner.nextLine();

System.out.print("Author: ");

String author2 = scanner.nextLine();

System.out.print("Page Count: ");

int pageCount2 = scanner.nextInt();

// Creating Book 2 using the constructor with all parameters

Book book2 = new Book(title2, author2, pageCount2);

// Displaying details of each book

System.out.println("Displaying Books Details");

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Book 1:");

book1.displayBookDetails();

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.out.println("Book 2:");

book2.displayBookDetails();

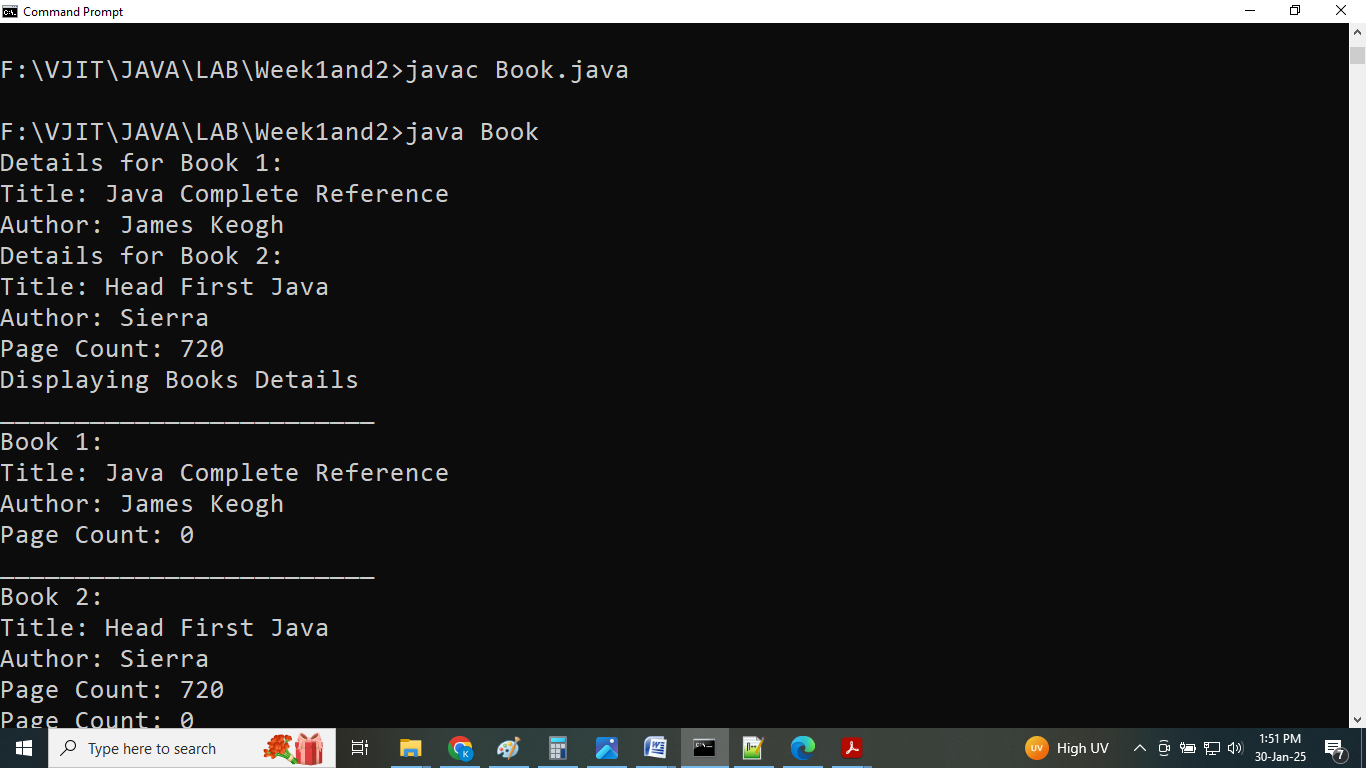
// Close the scanner

scanner.close();

}

}

**Output:**



##### **Write a program to illustrate method overloading.**

import java.util.Scanner;

public class **TestOverloading2**

{

public static void **main**(String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.print("Enter two integers for the first addition: ");

int num1 = scanner.nextInt();

int num2 = scanner.nextInt();

int intSum2 = sum(num1, num2); // Sum of two integers

System.out.println("Sum of "+num1+" and "+num2+": "+intSum2);

System.out.print("Enter three integers for addition: ");

int num3 = scanner.nextInt();

int num4 = scanner.nextInt();

int num5 = scanner.nextInt();

int intSum3 = sum(num3, num4, num5); // Sum of three integers

System.out.println("Sum of "+num3+","+num4+" and "+num5+": "+intSum3);

System.out.print("Enter two doubles for the second addition: ");

double num6 = scanner.nextDouble();

double num7 = scanner.nextDouble();

double doubleSum2 = sum(num6, num7); // Sum of two doubles

System.out.println("Sum of "+num6+" and "+num7+": "+doubleSum2);

System.out.print("Enter three doubles for the addition: ");

double num8 = scanner.nextDouble();

double num9 = scanner.nextDouble();

double num10 = scanner.nextDouble();

double doubleSum3 = sum(num8, num9, num10); // Sum of three doubles

System.out.println("Sum of "+num8+","+num9+" and "+num10+": "+doubleSum3);

scanner.close();

}

// Method to sum two integers

public static int **sum**(int a, int b)

{

return a + b;

}

// Method to sum three integers

public static int **sum**(int a, int b, int c)

{

return a + b + c;

}

// Method to sum two doubles

public static double **sum**(double a, double b)

{

return a + b;

}

// Method to sum three doubles

public static double **sum**(double a, double b, double c)

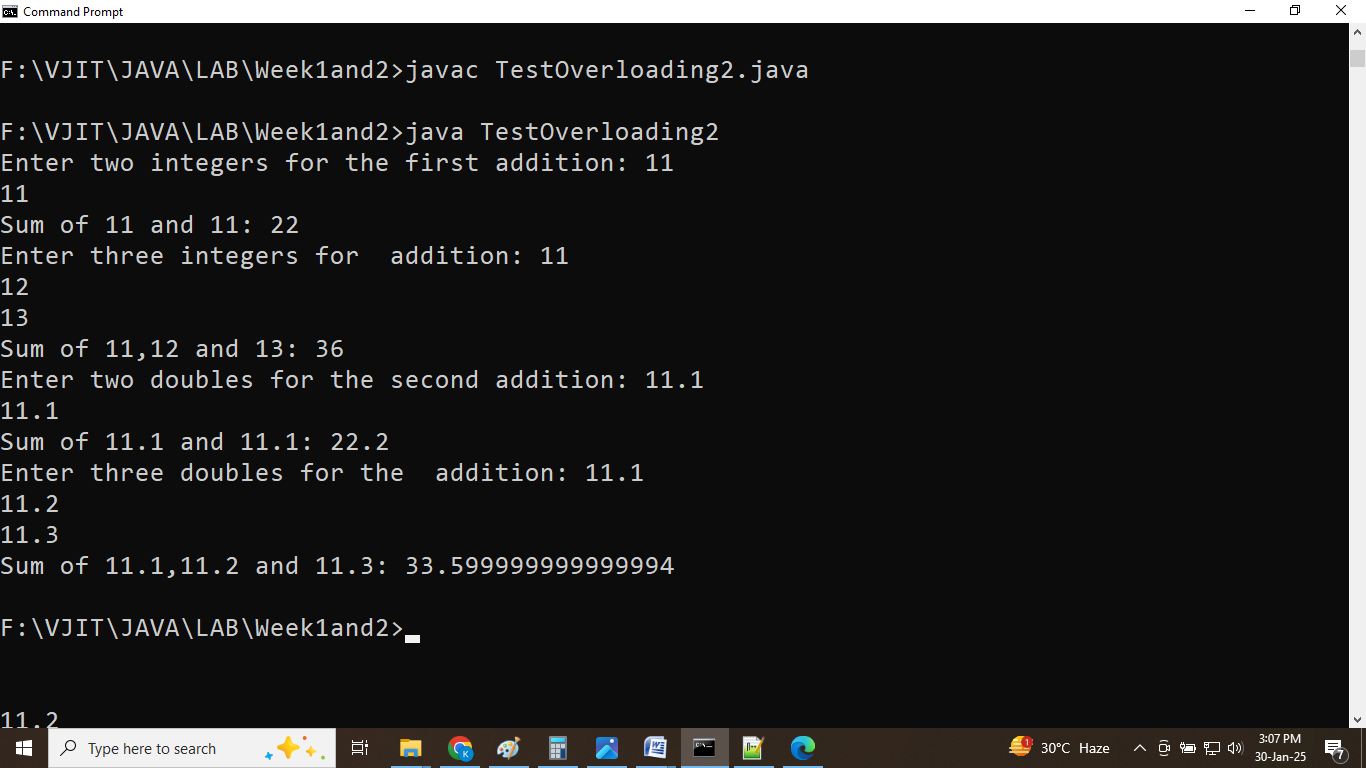
{

return a + b + c;

}

}

**Output:**



##### **Write a program to illustrate parameter passing using objects.**

import java.util.Scanner; // Import Scanner class for user input

// A class representing an Employee with name, department, and salary.

public class Employee

{

String name;

String department;

double salary;

// Constructor to initialize Employee object

Employee(String name, String department, double salary)

{

this.name = name;

this.department = department;

this.salary = salary;

}

// Method to display the Employee details on separate lines

void display()

{

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

System.out.println("Department: " + department);

System.out.println("Salary: Rs." + salary);

}

// Method to update the Employee's department

void updateDepartment(String newDepartment)

{

this.department = newDepartment;

}

// Method to update the Employee's salary

void updateSalary(double newSalary)

{

this.salary = newSalary;

}

}

class **EmployeeDetailsExample**

{

// A method that accepts an Employee object and modifies its details

public static void **updateEmployeeDetails**(Employee employee, String newDepartment, double newSalary) {

// Modifying the Employee object's fields

// Changing the department

employee.updateDepartment(newDepartment);

// Changing the salary

employee.updateSalary(newSalary);

}

public static void **main**(String[] args)

{

Scanner scanner = new Scanner(System.in);

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

String name = scanner.nextLine();

System.out.print("department : ");

String department = scanner.nextLine();

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

double salary = scanner.nextDouble();

// Create an Employee object using user input

Employee employee = new Employee(name, department, salary);

// Display the initial details of the employee

System.out.println("Before updateEmployeeDetails method call");

employee.display();

scanner.nextLine();

System.out.print("new department: ");

String newDepartment = scanner.nextLine();

System.out.print("new salary: ");

double newSalary = scanner.nextDouble();

// Update the Employee details with new department and salary

updateEmployeeDetails(employee, newDepartment, newSalary);

// Display the details of the employee after modification

System.out.println("After updateEmployeeDetails method call");

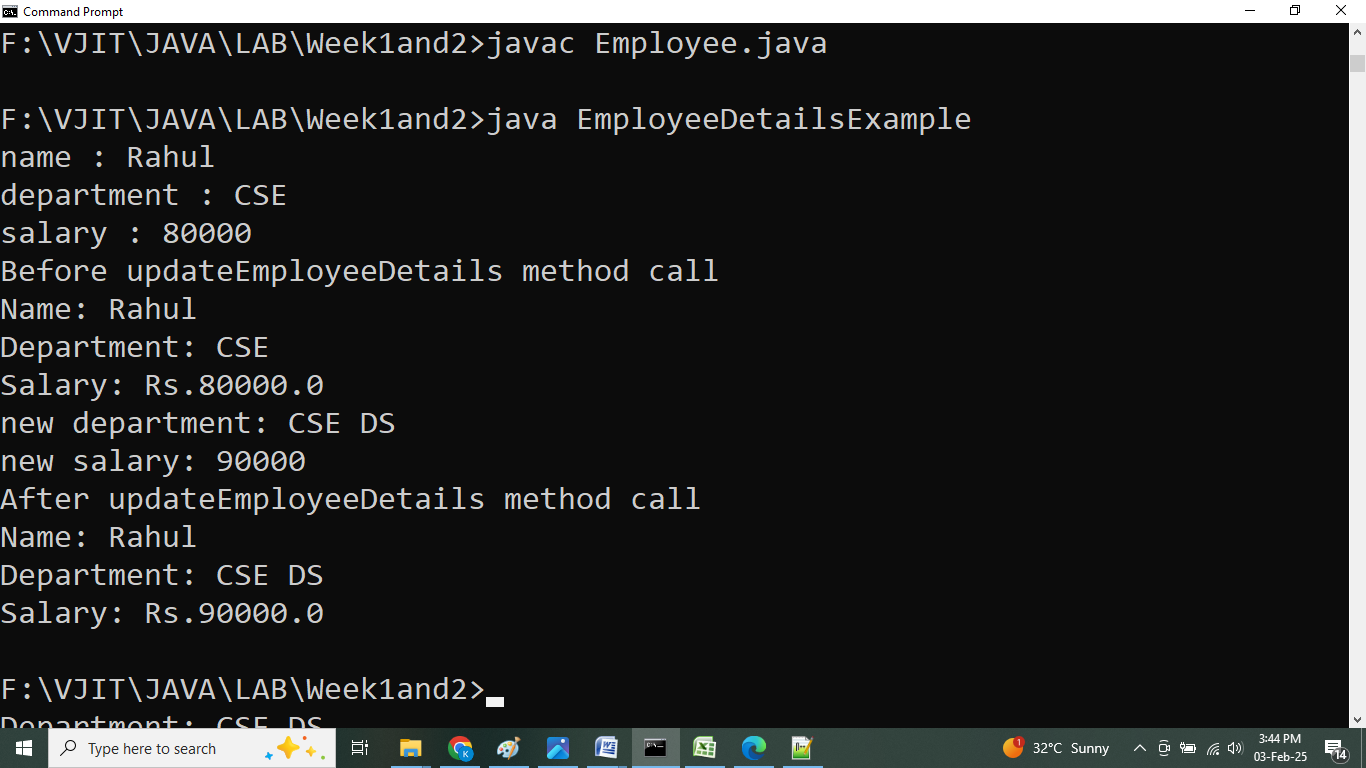
employee.display();

scanner.close();

}

}

**Output:**



##### **Write a program to illustrate Array Manipulation.**

import java.util.Scanner;

public class ArraySquaresPrinter

{

public static void main(String[] args)

{

// Create a Scanner object for input

Scanner sc = new Scanner(System.in);

// Read the size of the array

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

int N = sc.nextInt();

// Initialize an array to store the elements

int[] arr = new int[N];

System.out.println("Enter array elements");

// Read the array elements

for (int i = 0; i < N; i++)

{

arr[i] = sc.nextInt();

}

// Print the squares of the array elements

System.out.print("Squares of Array Elements are:");

for (int i = 0; i < N; i++)

{

System.out.print(arr[i] \* arr[i] + " ");

}

// Close the scanner

sc.close();

}

}

**Output:**

