**Program No. - 1**

**Program : Write a Java Program for Creation and Casting of Variables**

public class VariableCasting {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        System.out.println("Implicit casting (Winding conversion)\n");

        byte bytevar = 65;

        short shortvar = bytevar;

        System.out.println("bytevar : " + bytevar);

        System.out.println("Casting bytevar  to shortvar : " + shortvar + "\n");

        shortvar = 66;

        char charvar = (char)shortvar;

        System.out.println("shortvar : " + shortvar);

        System.out.println("Casting shortvar to charvar : " + charvar + "\n");

        charvar = 'A';

        int intvar = charvar;

        System.out.println("charvar : " + charvar);

        System.out.println("Casting charvar to intvar : " + intvar + "\n");

        intvar = 10;

        float floatvar = intvar;

        System.out.println("intvar : " + intvar);

        System.out.println("Casting intvar to floatvar : " + floatvar + "\n");

        floatvar = 10.1f;

        long longvar = (long)floatvar;

        System.out.println("floatvar : " + floatvar);

        System.out.println("Casting floatvar to longvar : " + longvar + "\n");

        longvar = 100l;

        double doublevar = longvar;

        System.out.println("longvar : " + longvar);

        System.out.println("Casting longvar to dobulevar : " + doublevar + "\n");

        System.out.println("Explicit casting (narrowing conversion)\n");

        doublevar = 5.5;

        longvar = (long)doublevar;

        System.out.println("doublevar : " + doublevar);

        System.out.println("Casting doublevar to longvar : " + longvar + "\n");

        longvar = 100l;

        floatvar = (float)longvar;

        System.out.println("longvar : " + longvar);

        System.out.println("Casting longvar to floatvar : " + floatvar + "\n");

        floatvar = 10.1f;

        intvar = (int)floatvar;

        System.out.println("floatvar : " + floatvar);

        System.out.println("Casting floatvar to intvar : " + intvar + "\n");

        intvar = 66;

        charvar = (char)intvar;

        System.out.println("intvar : " + intvar);

        System.out.println("Casting intvar to charvar : " + charvar + "\n");

        charvar = 'A';

        shortvar = (short)charvar;

        System.out.println("charvar : " + charvar);

        System.out.println("Casting charvar to shortvar : " + shortvar + "\n");

        shortvar = 66;

        bytevar = (byte)shortvar;

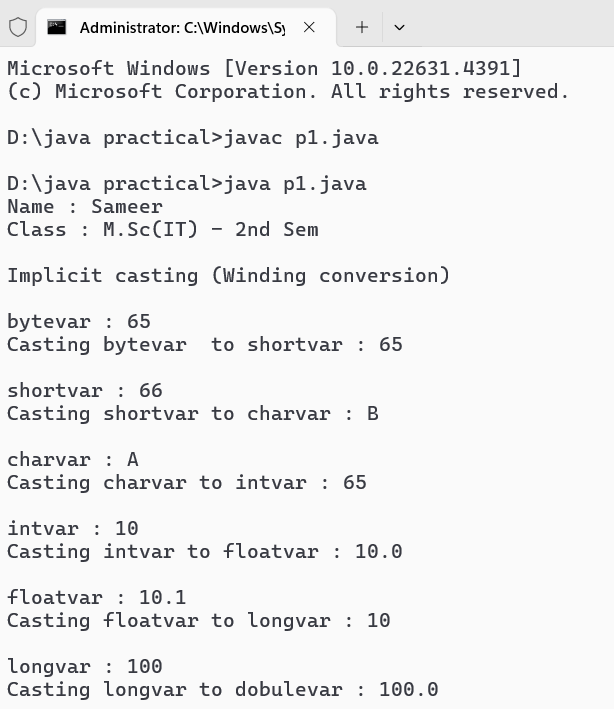
        System.out.println("shortvar : " + shortvar);

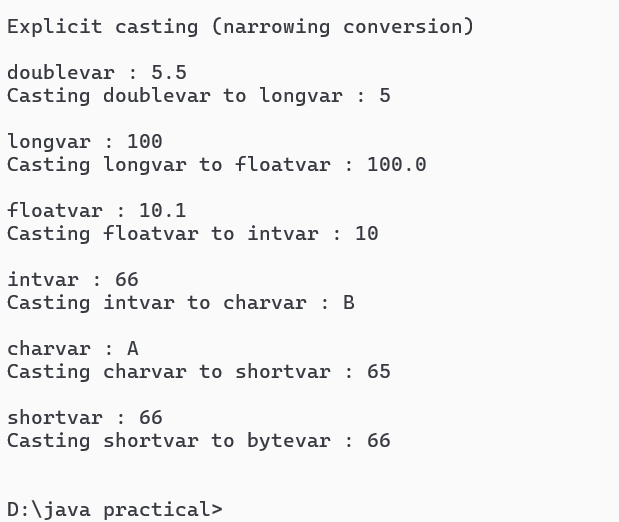
        System.out.println("Casting shortvar to bytevar : " + bytevar + "\n");

    }

}

**Output :-**

****

****

**Program No. - 2**

**WAJP to demonstrate the various Operators.**

public class Practical2 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

// Arithmetic Operators

int a = 10, b = 20;

System.out.println("Arithmetic Operators:");

System.out.println("a + b = " + (a + b)); // Addition

System.out.println("a - b = " + (a - b)); // Subtraction

System.out.println("a \* b = " + (a \* b)); // Multiplication

System.out.println("b / a = " + (b / a)); // Division

System.out.println("b % a = " + (b % a)); // Modulus

// Relational Operators

System.out.println("\nRelational Operators:");

System.out.println("a == b: " + (a == b)); // Equal to

System.out.println("a != b: " + (a != b)); // Not equal to

System.out.println("a > b: " + (a > b)); // Greater than

System.out.println("a < b: " + (a < b)); // Less than

System.out.println("a >= b: " + (a >= b)); // Greater than or equal to

System.out.println("a <= b: " + (a <= b)); // Less than or equal to

// Logical Operators

boolean x = true, y = false;

System.out.println("\nLogical Operators:");

System.out.println("x && y: " + (x && y)); // Logical AND

System.out.println("x || y: " + (x || y)); // Logical OR

System.out.println("!x: " + (!x)); // Logical NOT

// Bitwise Operators

System.out.println("\nBitwise Operators:");

System.out.println("a & b = " + (a & b)); // Bitwise AND

System.out.println("a | b = " + (a | b)); // Bitwise OR

System.out.println("a ^ b = " + (a ^ b)); // Bitwise XOR

System.out.println("~a = " + (~a)); // Bitwise NOT

System.out.println("a << 2 = " + (a << 2)); // Left shift

System.out.println("a >> 2 = " + (a >> 2)); // Right shift

// Assignment Operators

System.out.println("\nAssignment Operators:");

int c = 5;

System.out.println("c = " + c);

c += 3; // c = c + 3

System.out.println("c += 3: " + c);

c -= 2; // c = c - 2

System.out.println("c -= 2: " + c);

c \*= 2; // c = c \* 2

System.out.println("c \*= 2: " + c);

c /= 2; // c = c / 2

System.out.println("c /= 2: " + c);

c %= 2; // c = c % 2

System.out.println("c %= 2: " + c);

// Unary Operators

System.out.println("\nUnary Operators:");

int d = 10;

System.out.println("d = " + d);

System.out.println("++d = " + (++d)); // Pre-increment

System.out.println("d++ = " + (d++)); // Post-increment

System.out.println("d = " + d);

System.out.println("--d = " + (--d)); // Pre-decrement

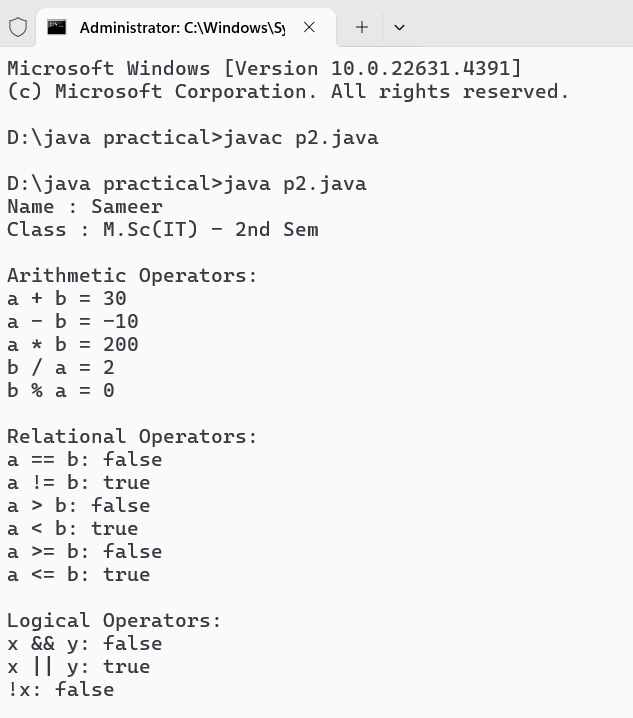
System.out.println("d-- = " + (d--)); // Post-decrement

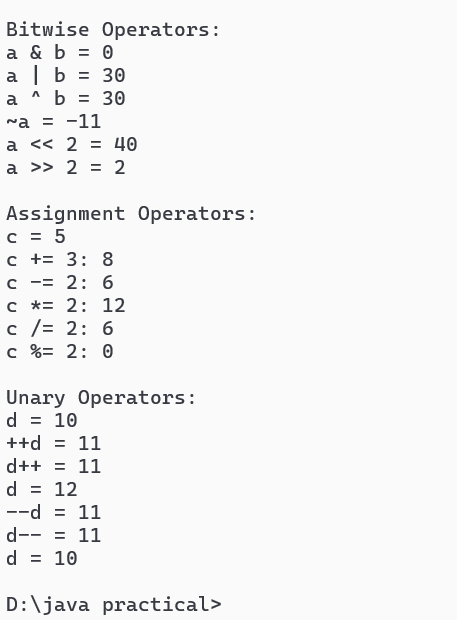
System.out.println("d = " + d);

}

}

**Output :-**





**Program No. - 3**

**WAJP for printing the current date in different formats.**

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

public class Practical3 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        // Get the current date

        LocalDate currentDate = LocalDate.now();

        // Define date formats

        DateTimeFormatter format1 = DateTimeFormatter.ofPattern("dd-MM-yyyy");

        DateTimeFormatter format2 = DateTimeFormatter.ofPattern("MM/dd/yyyy");

        DateTimeFormatter format3 = DateTimeFormatter.ofPattern("yyyy/MM/dd");

        DateTimeFormatter format4 = DateTimeFormatter.ofPattern("EEEE, MMMM dd, yyyy");

        // Print the current date in different formats

        System.out.println("Current date in different formats:");

        System.out.println("Format 1 (dd-MM-yyyy): " + currentDate.format(format1));

        System.out.println("Format 2 (MM/dd/yyyy): " + currentDate.format(format2));

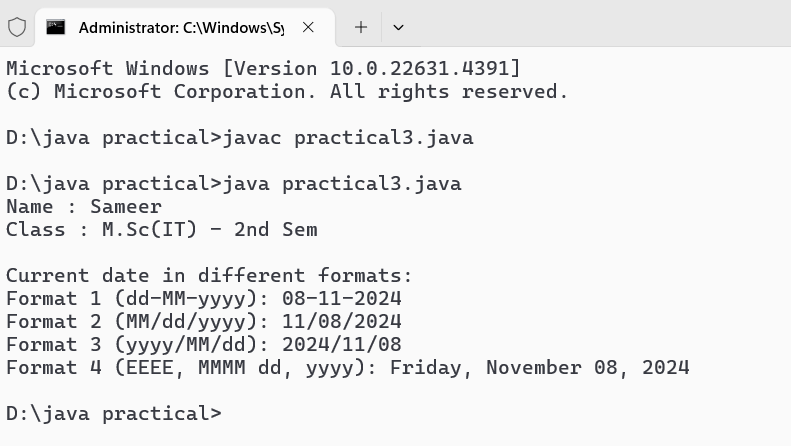
        System.out.println("Format 3 (yyyy/MM/dd): " + currentDate.format(format3));

        System.out.println("Format 4 (EEEE, MMMM dd, yyyy): " + currentDate.format(format4));

    }

}

**Output :-**



**Program No. - 4**

**WAJP for Inputting Data From Keyboard through Scanner Class.**

import java.util.Scanner;

public class Practical4 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

Scanner s1 = new Scanner(System.in);

int num;

String name;

System.out.println("Enter the student roll number : ");

num = s1.nextInt();

System.out.println("Enter the student name : ");

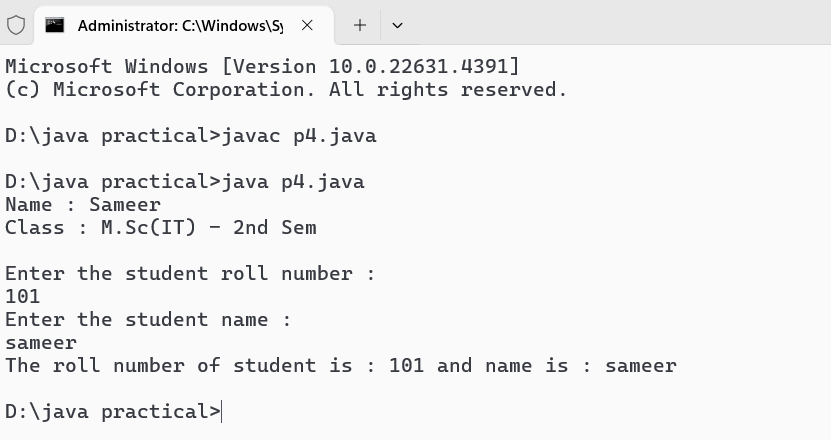
name = s1.next();

System.out.println("the student roll number is : " + num + " and Name is : " + name);

}

}

**Output :-**

****

**Program No. - 5**

**WAJP for Inputting Data From Keyboard through BufferedReader Class.**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Practical5 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Please enter some text:");

try {

String userInput = reader.readLine();

System.out.println("You entered: " + userInput);

} catch (IOException e) {

System.out.println("An error occurred while trying to read input: " + e.getMessage());

} finally {

try {

reader.close();

} catch (IOException e) {

System.out.println("Failed to close the BufferedReader: " + e.getMessage());

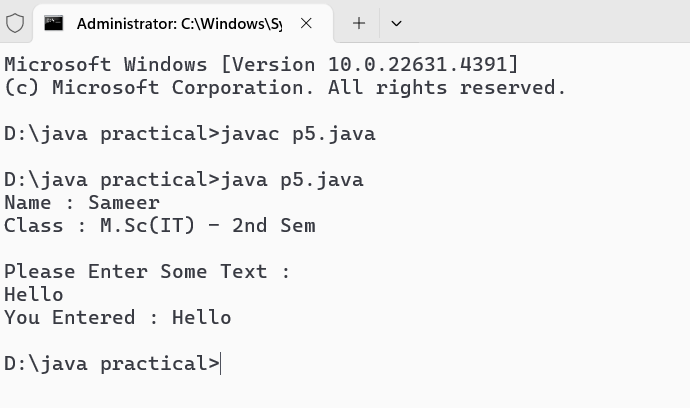
}

}

}

}

**Output :-**

****

**Program No. - 6**

**WAJP for Inputting Data From Keyboard through Console Class.**

import java.io.Console;

public class Practical6 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Console console = System.console();

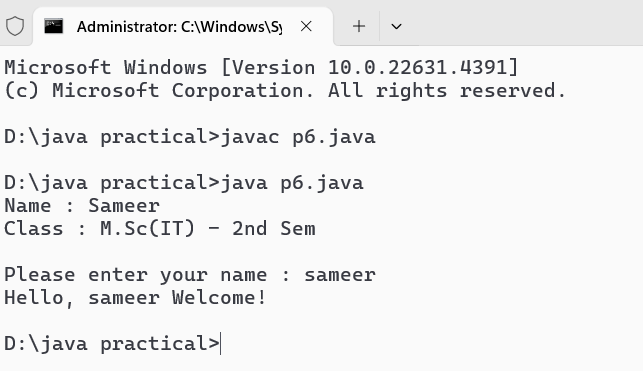
        String name = console.readLine("Please enter you name : ");

        console.printf("Hello, %s Welcome!\n", name);

    }

}

**Output :-**



**Program No. - 7**

**WAJP to demonstrate the use of for–each loop.**

public class Practical7 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

int[] numbers = {10, 20, 30, 40, 50};

System.out.println("Elements in the array:");

for (int number : numbers) {

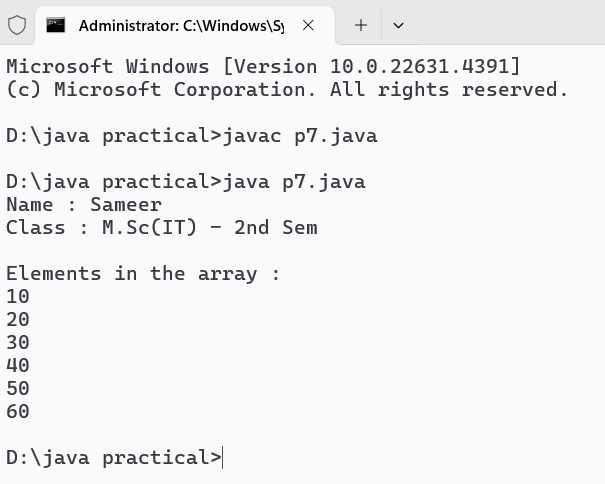
System.out.println(number);

}

}

}

**Output :-**

****

**Program No. - 8**

**WAJP to demonstrate ragged arrays.**

public class Main {

public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

// Declare a ragged array

int[][] raggedArray = new int[3][];

// Initialize the ragged array

raggedArray[0] = new int[3]; // first row has 3 elements

raggedArray[1] = new int[2]; // second row has 2 elements

raggedArray[2] = new int[4]; // third row has 4 elements

// Assign values to the ragged array

raggedArray[0][0] = 1;

raggedArray[0][1] = 2;

raggedArray[0][2] = 3;

raggedArray[1][0] = 4;

raggedArray[1][1] = 5;

raggedArray[2][0] = 6;

raggedArray[2][1] = 7;

raggedArray[2][2] = 8;

raggedArray[2][3] = 9;

// Print the ragged array

for (int i = 0; i < raggedArray.length; i++) {

for (int j = 0; j < raggedArray[i].length; j++) {

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

}

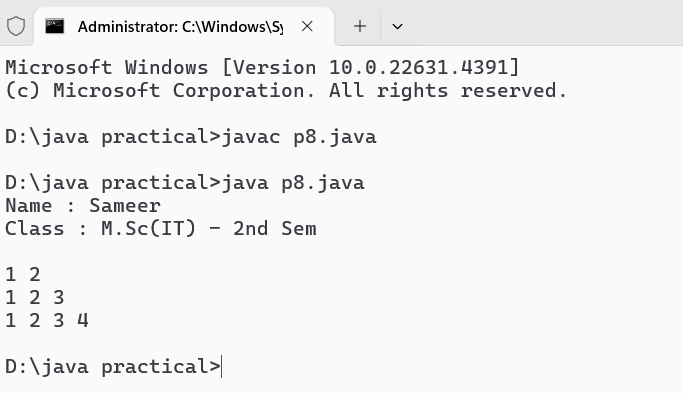
System.out.println();

}

}

}

**Output :-**

****

**Program No. - 9**

**WAJP to demonstrate anonymous arrays.**

public class Main {

// Method that takes an array as an argument

static void printArray(int[] array) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

for (int element : array) {

System.out.print(element + " ");

}

System.out.println();

}

public static void main(String[] args) {

// Creating and passing an anonymous array directly to the method

printArray(new int[]{1, 2, 3, 4, 5}); // Anonymous array with 5 elements

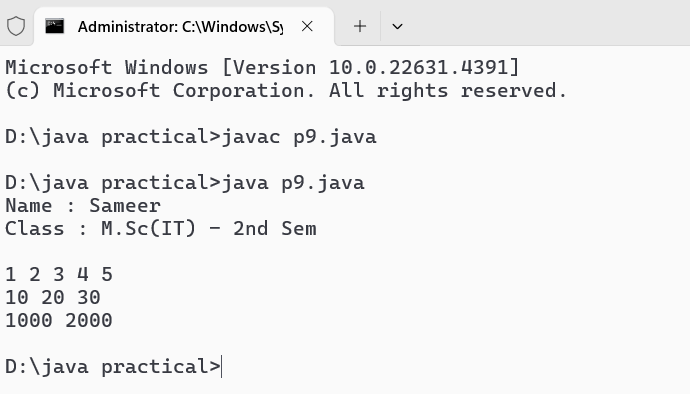
printArray(new int[]{10, 20, 30}); // Anonymous array with 3 elements

printArray(new int[]{100, 200}); // Anonymous array with 2 elements

}

}

**Output :-**

****

**Program No. - 10**

**WAJP to demonstrate the methods of Arrays Class.**

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

// Create an array

int[] numbers = {5, 3, 8, 1, 2};

// Print the original array

System.out.println("Original array: " + Arrays.toString(numbers));

// Sort the array

Arrays.sort(numbers);

System.out.println("Sorted array: " + Arrays.toString(numbers));

// Create another array for comparison

int[] numbers2 = {1, 2, 3, 4, 5};

// Check if two arrays are equal

boolean isEqual = Arrays.equals(numbers, numbers2);

System.out.println("Are the two arrays equal? " + isEqual);

// Fill an array with a specific value

int[] filledArray = new int[5];

Arrays.fill(filledArray, 7);

System.out.println("Filled array: " + Arrays.toString(filledArray));

// Copy an array

int[] copiedArray = Arrays.copyOf(numbers, numbers.length);

System.out.println("Copied array: " + Arrays.toString(copiedArray));

// Search for an element in the array

int index = Arrays.binarySearch(numbers, 3);

System.out.println("Index of element 3: " + index);

// Convert an array to a string

String[] stringArray = {"Java", "Python", "C++"};

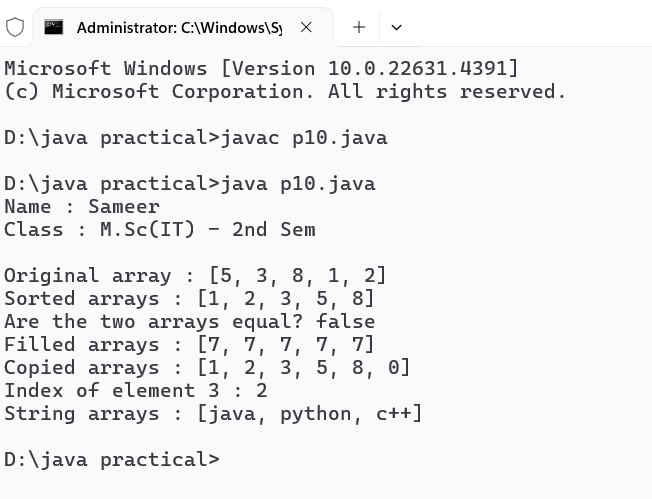
String arrayString = Arrays.toString(stringArray);

System.out.println("String array: " + arrayString);

}

}

**Output :-**

****

**Program No. - 11**

**WAJP for Application Of Classes And Objects**

class student{

    private String name;

    private int age;

    private double grades;

    public student(String name, int age, double grades){

        this.name = name;

        this.age = age;

        this.grades = grades;

    }

    public void displaydetails(){

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

        System.out.println("Student age : " + age);

        System.out.println("Student grade : " + grades);

    }

}

public class p11{

    public static void main(String[] args){

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        student stu1 = new student("Alice", 20,  85.5);

        student stu2 = new student("BoB", 22, 90.0);

        System.out.println("Details of student 1 : ");

        stu1.displaydetails();

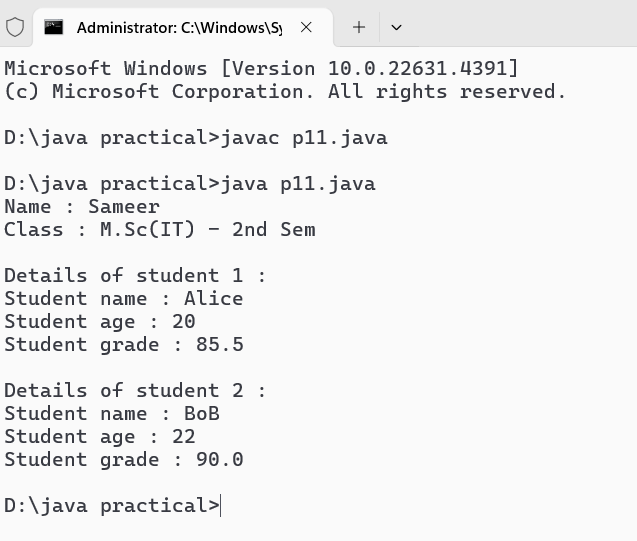
        System.out.println("\nDetails of student 2 : ");

        stu2.displaydetails();

    }

}

**Output :-**



**Program No. - 12**

**WAJP to demonstrate method overloading.**

class methoperations{

    public int add(int a, int b){

        return a + b;

    }

    public int add(int a, int b, int c){

        return a + b + c;

    }

    public double add(double a, double b){

        return a + b;

    }

    public String add(String a, String b){

        return a + b;

    }

}

public class p12{

    public static void main(String[] args){

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        methoperations mathops = new methoperations();

        System.out.println("Sum of 5 and 10 : " + mathops.add(5, 10));

        System.out.println("Sum of 5, 10 and 15 : " + mathops.add(5, 10, 15));

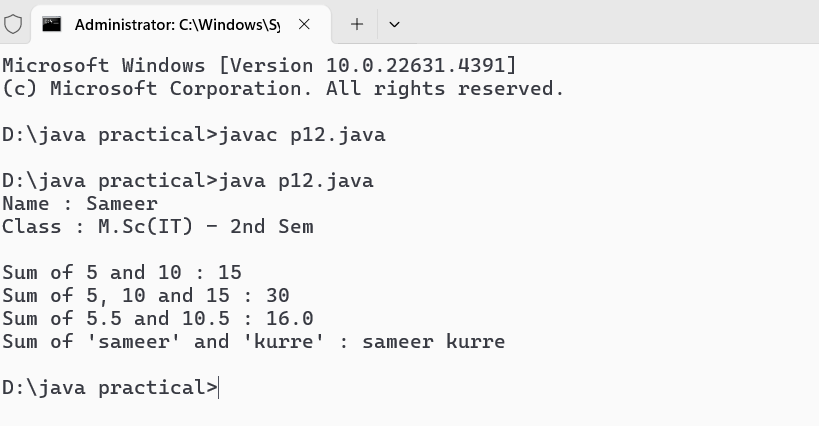
        System.out.println("Sum of 5.5 and 10.5 : " + mathops.add(5.5, 10.5));

        System.out.println("Sum of 'sameer' and ' kurre' : " + mathops.add("sameer", " kurre"));

    }

}

**Output :-**



**Program No. - 13**

**WAJP to demonstrate constructor overloading.**

class Book {

    private String title;

    private String author;

    private double price;

    public Book() {

        this.title = "Unknown Title";

        this.author = "Unknown Author";

        this.price = 0.0;

    }

    public Book(String title) {

        this.title = title;

        this.author = "Unknown Author";

        this.price = 0.0;

    }

    public Book(String title, String author) {

        this.title = title;

        this.author = author;

        this.price = 0.0;

    }

    public Book(String title, String author, double price) {

        this.title = title;

        this.author = author;

        this.price = price;

    }

    public void display() {

        System.out.println("Title: " + title + ", Author: " + author + ", Price: " + price);

    }

}

public class p13 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Book book1 = new Book();

        System.out.println("\nDefault Constructor");

        book1.display();

        System.out.println("\n1 Argument Constructor");

        Book book2 = new Book("1984");

        book2.display();

        System.out.println("\n2 Argument Constructor");

        Book book3 = new Book("To Kill a Mockingbird", "Harper Lee");

        book3.display();

        System.out.println("\n3 Argument Constructor");

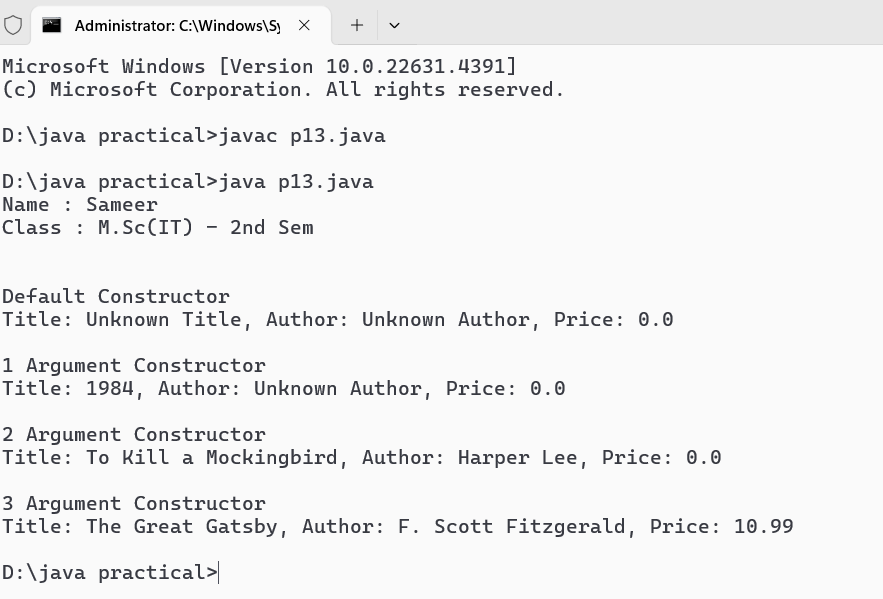
        Book book4 = new Book("The Great Gatsby", "F. Scott Fitzgerald", 10.99);

        book4.display();

    }

}

**Output :-**



**Program No. - 14**

**WAJP Using Single Inheritance.**

// Superclass

class Animal {

    void eat() {

        System.out.println("This animal eats food.");

    }

}

// Subclass

class Dog extends Animal {

    void bark() {

        System.out.println("The dog barks.");

    }

}

// Driver class

public class p14 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Dog myDog = new Dog();

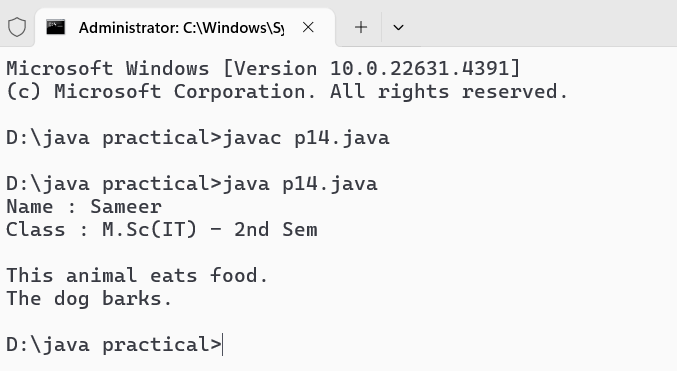
        myDog.eat();

        myDog.bark();

    }

}

**Output :-**



**Program No. - 15**

**WAJP Using Super And This Keyword.**

class Animal {

    String name;

    Animal(String name) {

        this.name = name;

    }

    void display() {

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

    }

}

class Dog extends Animal {

    String name;

    Dog(String animalName, String dogName) {

        super(animalName);

        this.name = dogName;

    }

    void display() {

        super.display();

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

    }

}

public class p15 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

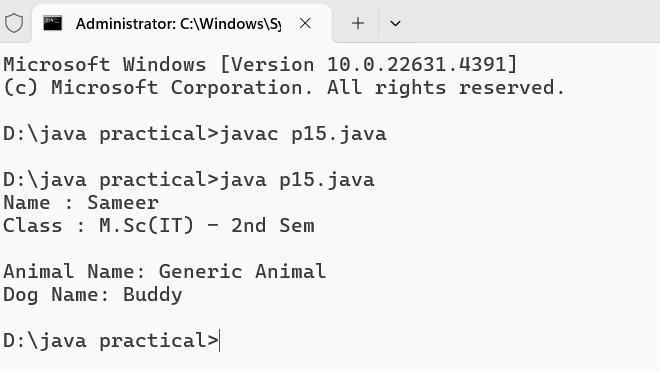
        Dog dog = new Dog("Generic Animal", "Buddy");

        dog.display();

    }

}

**Output :-**



**Program No. - 16**

**WAJP to demonstrate multilevel inheritance.**

class Animal {

    String name;

    public Animal(String name) {

        this.name = name;

    }

    public String speak() {

        return "Some sound";

    }

}

class Mammal extends Animal {

    String furColor;

    public Mammal(String name, String furColor) {

        super(name);

        this.furColor = furColor;

    }

    @Override

    public String speak() {

        return "Mammal sound";

    }

}

class Dog extends Mammal {

    String breed;

    public Dog(String name, String furColor, String breed) {

        super(name, furColor);

        this.breed = breed;

    }

    @Override

    public String speak() {

        return "Woof!";

    }

}

public class p16 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Dog myDog = new Dog("Buddy", "Brown", "Golden Retriever");

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

        System.out.println("Fur Color: " + myDog.furColor);

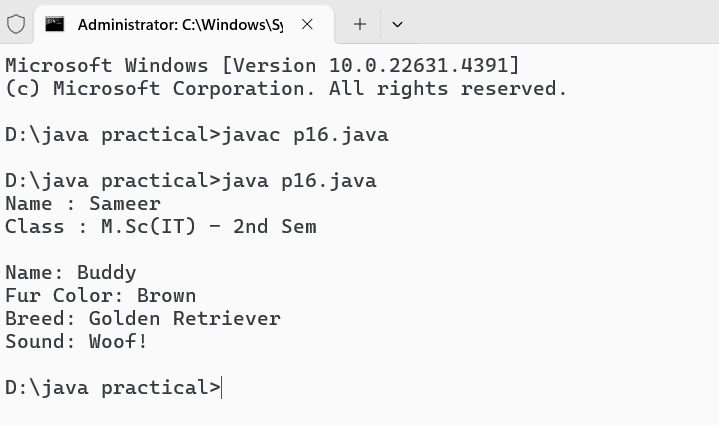
        System.out.println("Breed: " + myDog.breed);

        System.out.println("Sound: " + myDog.speak());

    }

}

**Output :-**



**Program No. - 17**

**WAJP to demonstrate method overriding.**

class Animal {

    void sound() {

        System.out.println("Animal makes a sound");

    }

}

class Dog extends Animal {

    @Override

    void sound() {

        System.out.println("Dog barks");

    }

}

class Cat extends Animal {

    @Override

    void sound() {

        System.out.println("Cat meows");

    }

}

public class p17 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Animal myAnimal = new Animal();

        Animal myDog = new Dog();

        Animal myCat = new Cat();

        myAnimal.sound();

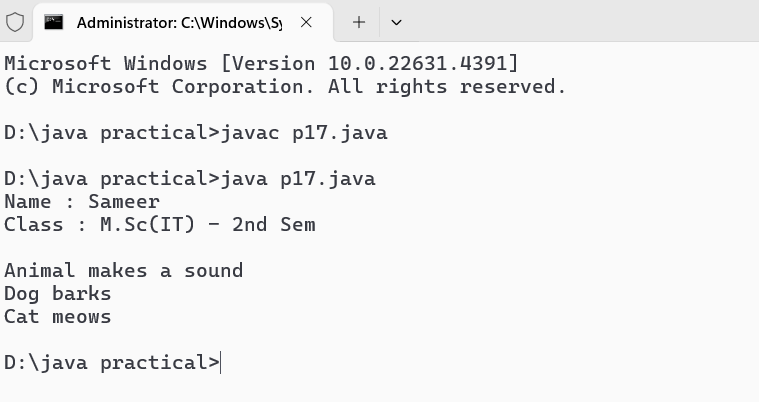
        myDog.sound();

        myCat.sound();

    }

}

**Output :-**

****

**Program No. - 18**

**WAJP Using Multiple Inheritance Concept through interfaces.**

interface Animal {

    void eat();

}

interface Pet {

    void play();

}

class Dog implements Animal, Pet {

    @Override

    public void eat() {

        System.out.println("Dog is eating.");

    }

    @Override

    public void play() {

        System.out.println("Dog is playing.");

    }

}

public class p18 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Dog myDog = new Dog();

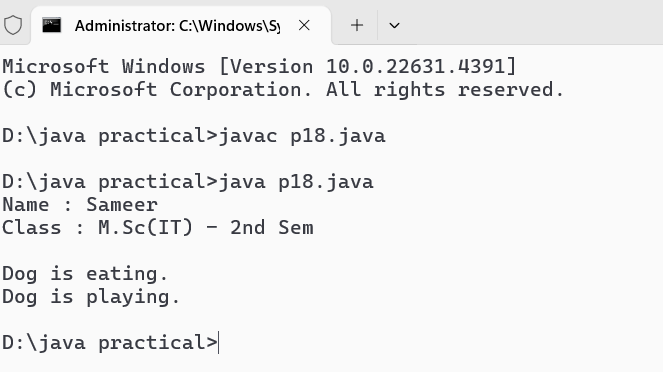
        myDog.eat();

        myDog.play();

    }

}

**Output :-**



**Program No. - 19**

**WAJP to demonstrate the concept of inner class.**

class OuterClass {

    private String outerField = "Outer field value";

    class InnerClass {

        void display() {

            System.out.println("Accessing: " + outerField);

        }

    }

    void createInnerClassInstance() {

        InnerClass inner = new InnerClass();

        inner.display();

    }

}

public class p19 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

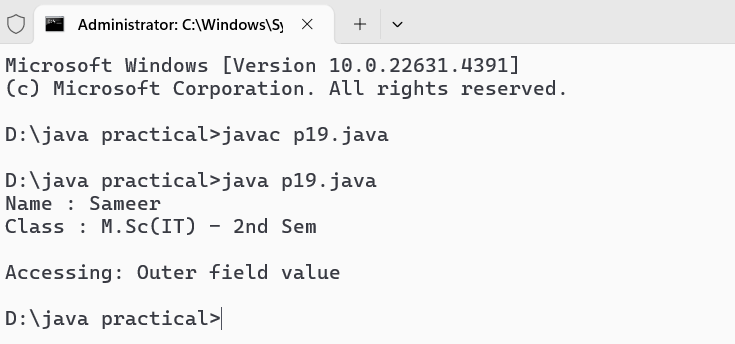
        OuterClass outer = new OuterClass();

        outer.createInnerClassInstance();

    }

}

**Output :-**



**Program No. - 20**

**WAJP to demonstrate the concept of local class.**

class OuterClass {

    private String outerField = "Outer field value";

    void outerMethod() {

        class LocalClass {

            void display() {

                System.out.println("Accessing: " + outerField);

            }

        }

        LocalClass localClassInstance = new LocalClass();

        localClassInstance.display();

    }

}

public class p20 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

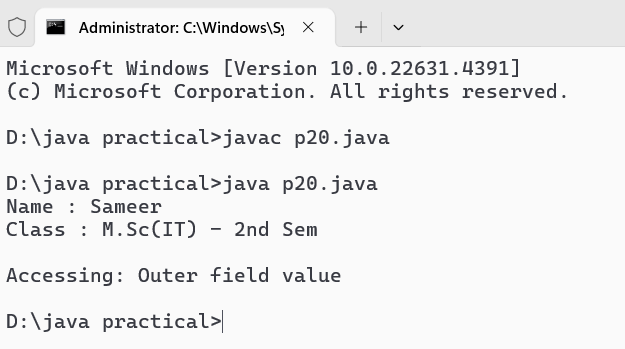
        OuterClass outer = new OuterClass();

        outer.outerMethod();

    }

}

**Output :-**

****

**Program No. - 21**

**WAJP that creates its own package containing two classes.**

**Program No. - 22.**

**WAJP Using Try And Catch Statement.**

import java.util.Scanner;

public class p22 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter the numerator: ");

            int numerator = scanner.nextInt();

            System.out.print("Enter the denominator: ");

            int denominator = scanner.nextInt();

            int result = numerator / denominator;

            System.out.println("Result: " + result);

        } catch (ArithmeticException e) {

            System.out.println("Error: Cannot divide by zero.");

        } catch (Exception e) {

            System.out.println("Error: " + e.getMessage());

        } finally {

            scanner.close();

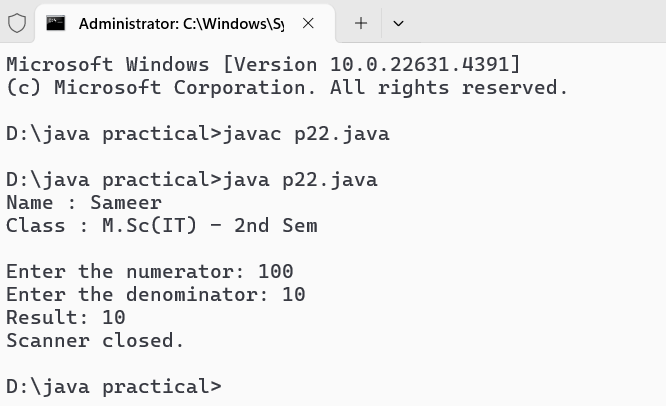
            System.out.println("Scanner closed.");

        }

    }

}

**Output :-**



**Program No. - 23**

**WAJP Using Multiple Catch Statements.**

import java.util.Scanner;

public class p23 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter the numerator: ");

            int numerator = scanner.nextInt();

            System.out.print("Enter the denominator: ");

            int denominator = scanner.nextInt();

            // Perform division

            int result = numerator / denominator;

            System.out.println("Result: " + result);

        } catch (ArithmeticException e) {

            System.out.println("Error: Cannot divide by zero.");

        } catch (java.util.InputMismatchException e) {

            System.out.println("Error: Invalid input. Please enter integers only.");

        } catch (Exception e) {

            System.out.println("An unexpected error occurred: " + e.getMessage());

        } finally {

            scanner.close();

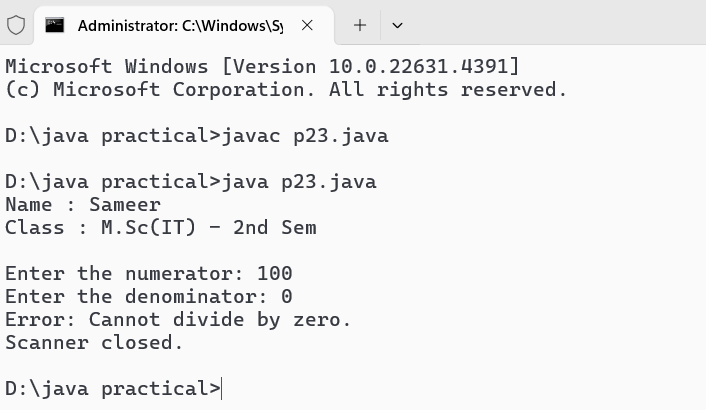
            System.out.println("Scanner closed.");

        }

    }

}

**Output :-**



**Program No. - 24**

**WAJP to demonstrate the MultiCatch feature.**

import java.util.Scanner;

public class p24 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter the numerator: ");

            int numerator = scanner.nextInt();

            System.out.print("Enter the denominator: ");

            int denominator = scanner.nextInt();

            // Perform division

            int result = numerator / denominator;

            System.out.println("Result: " + result);

        } catch (ArithmeticException | java.util.InputMismatchException e) {

            if (e instanceof ArithmeticException) {

                System.out.println("Error: Cannot divide by zero.");

            } else {

                System.out.println("Error: Invalid input. Please enter integers only.");

            }

        } catch (Exception e) {

            System.out.println("An unexpected error occurred: " + e.getMessage());

        } finally {

            scanner.close();

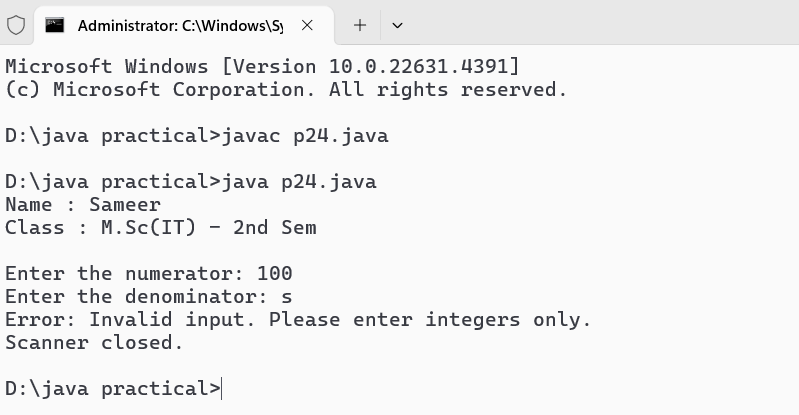
            System.out.println("Scanner closed.");

        }

    }

}

**Output :-**



**Program No. - 25**

**WAJP to demonstrate the use of finally block.**

import java.util.Scanner;

public class p25 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter the numerator: ");

            int numerator = scanner.nextInt();

            System.out.print("Enter the denominator: ");

            int denominator = scanner.nextInt();

            // Perform division

            int result = numerator / denominator;

            System.out.println("Result: " + result);

        } catch (ArithmeticException e) {

            System.out.println("Error: Cannot divide by zero.");

        } catch (java.util.InputMismatchException e) {

            System.out.println("Error: Invalid input. Please enter integers only.");

        } catch (Exception e) {

            System.out.println("An unexpected error occurred: " + e.getMessage());

        } finally {

            // This block will always execute

            scanner.close();

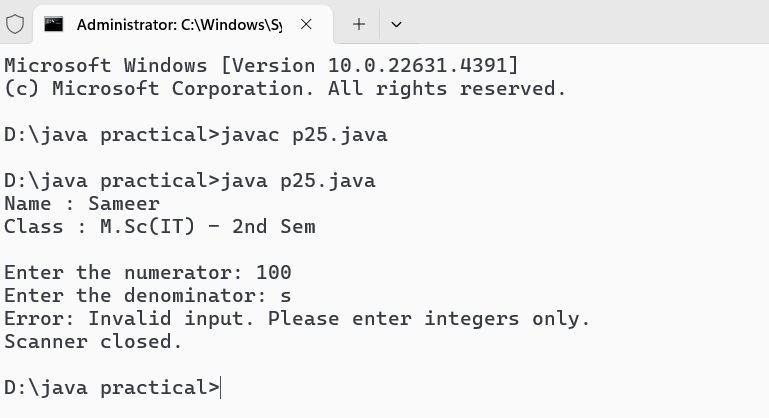
            System.out.println("Scanner closed.");

        }

    }

}

**Output :-**



**Program No. - 26**

**WAJP Using Nested Try Statements.**

import java.util.Scanner;

public class p26 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter the numerator: ");

            int numerator = scanner.nextInt();

            try {

                System.out.print("Enter the denominator: ");

                int denominator = scanner.nextInt();

                // Perform division

                int result = numerator / denominator;

                System.out.println("Result: " + result);

            } catch (ArithmeticException e) {

                System.out.println("Error: Cannot divide by zero.");

            } catch (java.util.InputMismatchException e) {

                System.out.println("Error: Invalid input for denominator. Please enter an integer.");

            }

        } catch (java.util.InputMismatchException e) {

            System.out.println("Error: Invalid input for numerator. Please enter an integer.");

        } finally {

            scanner.close();

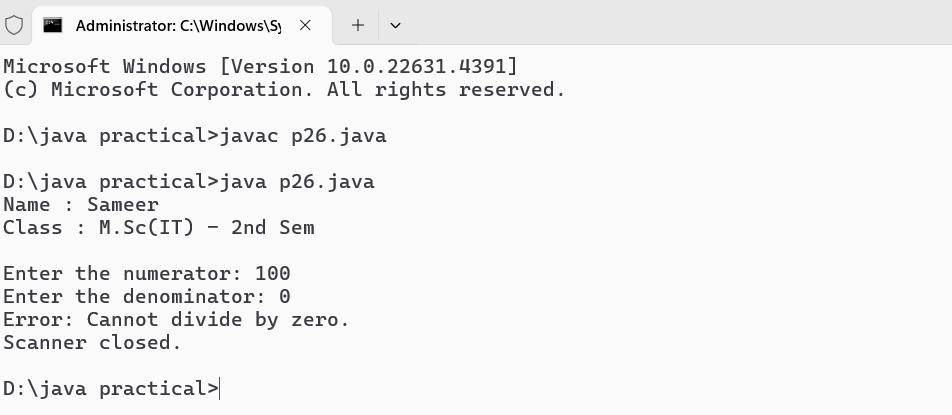
            System.out.println("Scanner closed.");

        }

    }

}

**Output :-**

****

**Program No. - 27**

**WAJP To Create Your Own Exception Class And Display Corresponding Error Message.**

import java.util.Scanner;

class InvalidAgeException extends Exception {

    public InvalidAgeException(String message) {

        super(message);

    }

}

public class p27 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        try {

            int age = getAge();

            System.out.println("Your age is: " + age);

        } catch (InvalidAgeException e) {

            System.out.println("Error: " + e.getMessage());

        }

    }

        public static int getAge() throws InvalidAgeException {

        Scanner scanner = new Scanner(System.in);

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

        int age = scanner.nextInt();

        if (age < 0 || age > 150) {

            throw new InvalidAgeException("Age must be between 0 and 150.");

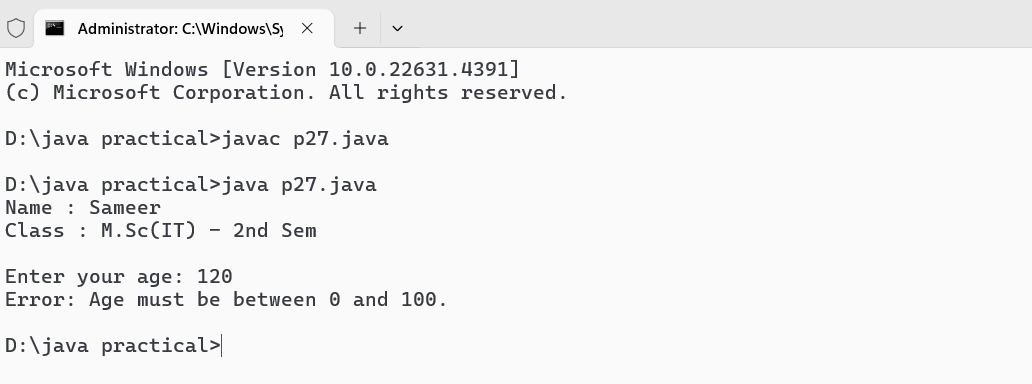
        }

        return age;

    }

}

**Output :-**



**Program No. - 28**

**WAJP For Creating And Executing Threads by extending the Thread class.**

class MyThread extends Thread {

    private String threadName;

    public MyThread(String name) {

        this.threadName = name;

    }

    @Override

    public void run() {

        for (int i = 0; i < 5; i++) {

            System.out.println(threadName + " - Count: " + i);

            try {

                Thread.sleep(500);

            } catch (InterruptedException e) {

                System.out.println(threadName + " interrupted.");

            }

        }

        System.out.println(threadName + " has finished executing.");

    }

}

public class p28 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        MyThread thread1 = new MyThread("Thread-1");

        MyThread thread2 = new MyThread("Thread-2");

        thread1.start();

        thread2.start();

        try {

            thread1.join();

            thread2.join();

        } catch (InterruptedException e) {

            System.out.println("Main thread interrupted.");

        }

        System.out.println("All threads have finished executing.");

    }

}

**Output :-**

A screenshot of a computer

Description automatically generated

**Program No. - 29**

**WAJP To run Three Threads by implementing the Runnable Interface.**

class MyRunnable implements Runnable {

    private String threadName;

    public MyRunnable(String name) {

        this.threadName = name;

    }

    @Override

    public void run() {

        for (int i = 0; i < 5; i++) {

            System.out.println(threadName + " - Count: " + i);

            try {

                Thread.sleep(500);

            } catch (InterruptedException e) {

                System.out.println(threadName + " interrupted.");

            }

        }

        System.out.println(threadName + " has finished executing.");

    }

}

public class p29 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        MyRunnable runnable1 = new MyRunnable("Thread-1");

        MyRunnable runnable2 = new MyRunnable("Thread-2");

        MyRunnable runnable3 = new MyRunnable("Thread-3");

        Thread thread1 = new Thread(runnable1);

        Thread thread2 = new Thread(runnable2);

        Thread thread3 = new Thread(runnable3);

        thread1.start();

        thread2.start();

        thread3.start();

        try {

            thread1.join();

            thread2.join();

            thread3.join();

        } catch (InterruptedException e) {

            System.out.println("Main thread interrupted.");

        }

        System.out.println("All threads have finished executing.");

    }

}

**Output :-**

A screenshot of a computer program

Description automatically generated

**Program No. - 30**

**WAJP to demonstrate the use of join() method.**

class MyThread extends Thread {

    private String threadName;

    MyThread(String name) {

        threadName = name;

    }

    public void run() {

        System.out.println(threadName + " is starting.");

        try {

            for (int i = 0; i < 5; i++) {

                System.out.println(threadName + " is working... " + i);

                Thread.sleep(1000);

            }

        } catch (InterruptedException e) {

            System.out.println(threadName + " was interrupted.");

        }

        System.out.println(threadName + " is finished.");

    }

}

public class p30 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        MyThread thread1 = new MyThread("Thread 1");

        MyThread thread2 = new MyThread("Thread 2");

        thread1.start();

        thread2.start();

        try {

            thread1.join();

            System.out.println("Thread 1 has finished. Now proceeding to Thread 2.");

            thread2.join();

            System.out.println("Thread 2 has finished.");

        } catch (InterruptedException e) {

            System.out.println("Main thread was interrupted.");

        }

        System.out.println("All threads have completed.");

    }

}

**Output :-**

**A screenshot of a computer program

Description automatically generated**

**Program No. - 31**

**WAJP to demonstrate Multithreading using wait () & notify().**

**class SharedResource {**

**private int data;**

**private boolean available = false;**

**public synchronized void produce(int value) {**

**while (available) {**

**try { wait(); } catch (InterruptedException e) { Thread.currentThread().interrupt(); }**

**}**

**data = value;**

**System.out.println("Produced: " + data);**

**available = true;**

**notify();**

**}**

**public synchronized int consume() {**

**while (!available) {**

**try { wait(); } catch (InterruptedException e) { Thread.currentThread().interrupt(); }**

**}**

**System.out.println("Consumed: " + data);**

**available = false;**

**notify();**

**return data;**

**}**

**}**

**class Producer extends Thread {**

**private final SharedResource resource;**

**public Producer(SharedResource resource) { this.resource = resource; }**

**public void run() {**

**for (int i = 0; i < 5; i++) {**

**resource.produce(i);**

**try { Thread.sleep(500); } catch (InterruptedException e) { Thread.currentThread().interrupt(); }**

**}**

**}**

**}**

**class Consumer extends Thread {**

**private final SharedResource resource;**

**public Consumer(SharedResource resource) { this.resource = resource; }**

**public void run() {**

**for (int i = 0; i < 5; i++) {**

**resource.consume();**

**try { Thread.sleep(1000); } catch (InterruptedException e) { Thread.currentThread().interrupt(); }**

**}**

**}**

**}**

**public class p31 {**

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

**System.out.println("Name : Sameer");**

**System.out.println("Class : M.Sc(IT) - 3rdSem\n");**

**SharedResource resource = new SharedResource();**

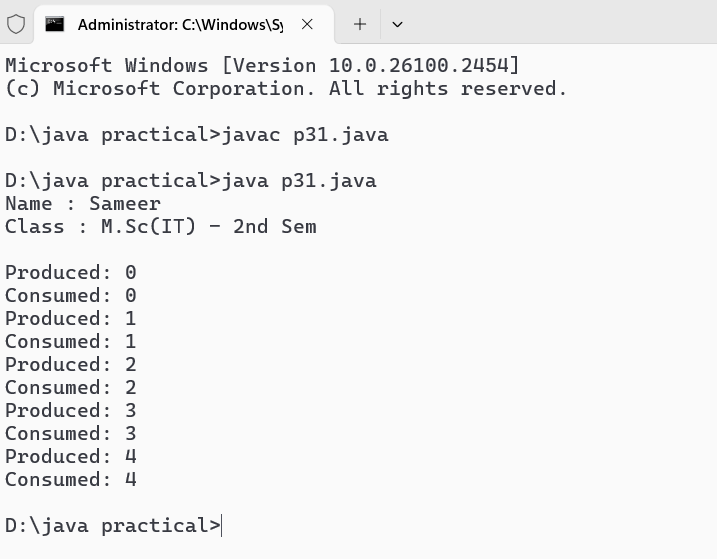
**new Producer(resource).start();**

**new Consumer(resource).start();**

**}**

**}**

**Output :-**



**Program No. - 32**

**WAJP to demonstrate The String Class & its methods.**

public class p32 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        String str1 = "Hello, World!";

        String str2 = "Java Programming";

        String str3 = "hello, world!";

        System.out.println("Original String 1: " + str1);

        System.out.println("Original String 2: " + str2);

        System.out.println("Original String 3: " + str3);

        System.out.println("Length of str1: " + str1.length());

        System.out.println("str1 in uppercase: " + str1.toUpperCase());

        System.out.println("str2 in lowercase: " + str2.toLowerCase());

        System.out.println("str1 equals str3: " + str1.equals(str3));

        System.out.println("str1 equals str3 (ignore case): " + str1.equalsIgnoreCase(str3));

        System.out.println("str2 contains 'Programming': " + str2.contains("Programming"));

        System.out.println("Index of 'W' in str1: " + str1.indexOf('W'));

        System.out.println("Index of 'World' in str1: " + str1.indexOf("World"));

        System.out.println("Substring of str2 (from index 5): " + str2.substring(5));

        System.out.println("str1 after replacing 'World' with 'Java': " + str1.replace("World", "Java"));

        String[] words = str1.split(" ");

        System.out.println("Words in str1:");

        for (String word : words) {

            System.out.println(word);

        }

        String str4 = "   Hello, Java!   ";

        System.out.println("Trimmed str4: '" + str4.trim() + "'");

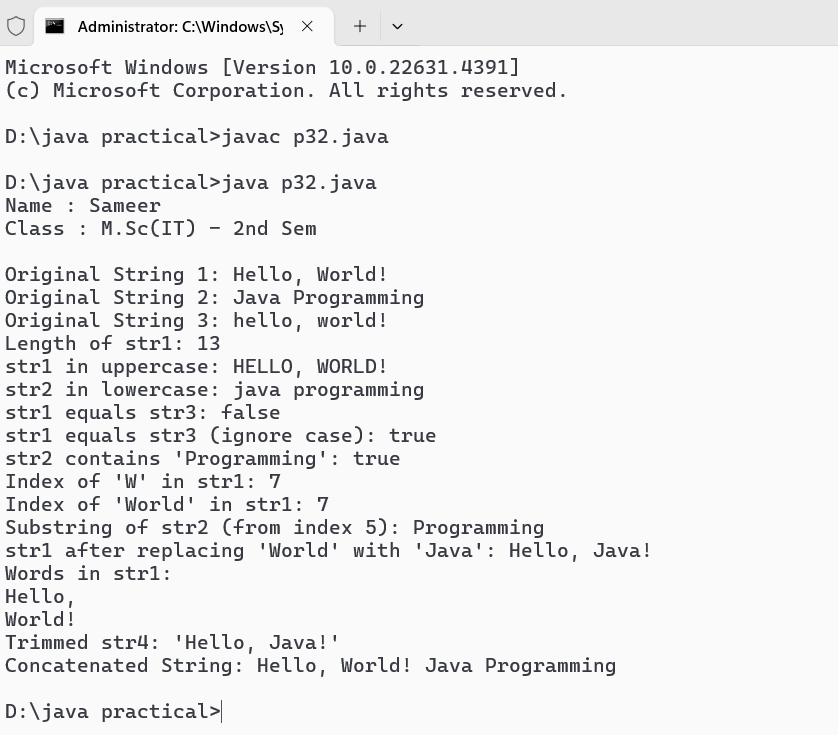
        String str5 = str1 + " " + str2;

        System.out.println("Concatenated String: " + str5);

    }

}

**Output :-**



**Program No. - 33**

**WAJP to demonstrate StringBuffer Class & its methods.**

public class p33 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        StringBuffer sb = new StringBuffer("Hello");

        System.out.println("Original StringBuffer: " + sb);

        sb.append(", World!");

        System.out.println("After append: " + sb);

        sb.insert(5, " Java");

        System.out.println("After insert: " + sb);

        sb.reverse();

        System.out.println("After reverse: " + sb);

        sb.reverse();

        System.out.println("Reversed back: " + sb);

        sb.replace(5, 10, " Everyone");

        System.out.println("After replace: " + sb);

        sb.delete(5, 14);

        System.out.println("After delete: " + sb);

        System.out.println("Length of StringBuffer: " + sb.length());

        System.out.println("Capacity of StringBuffer: " + sb.capacity());

        sb.ensureCapacity(50);

        System.out.println("New capacity after ensuring: " + sb.capacity());

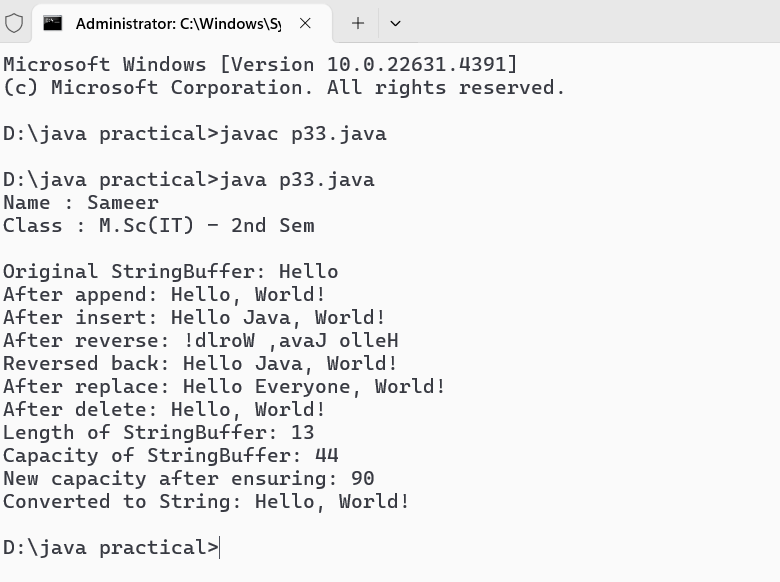
        String str = sb.toString();

        System.out.println("Converted to String: " + str);

    }

}

**Output :-**



**Program No. - 34**

**WAJP to demonstrate various Wrapper Classes.**

public class p34 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        int intValue = 42;

        Integer integerObject = Integer.valueOf(intValue);

        System.out.println("Integer Object: " + integerObject);

        int unboxedInt = integerObject.intValue();

        System.out.println("Unboxed Integer: " + unboxedInt);

        double doubleValue = 3.14;

        Double doubleObject = Double.valueOf(doubleValue);

        System.out.println("Double Object: " + doubleObject);

        double unboxedDouble = doubleObject.doubleValue();

        System.out.println("Unboxed Double: " + unboxedDouble);

        char charValue = 'A';

        Character characterObject = Character.valueOf(charValue);

        System.out.println("Character Object: " + characterObject);

        char unboxedChar = characterObject.charValue();

        System.out.println("Unboxed Character: " + unboxedChar);

        boolean boolValue = true;

        Boolean booleanObject = Boolean.valueOf(boolValue);

        System.out.println("Boolean Object: " + booleanObject);

        boolean unboxedBool = booleanObject.booleanValue();

        System.out.println("Unboxed Boolean: " + unboxedBool);

        long longValue = 123456789L;

        Long longObject = Long.valueOf(longValue);

        System.out.println("Long Object: " + longObject);

        long unboxedLong = longObject.longValue();

        System.out.println("Unboxed Long: " + unboxedLong);

        float floatValue = 5.75f;

        Float floatObject = Float.valueOf(floatValue);

        System.out.println("Float Object: " + floatObject);

        float unboxedFloat = floatObject.floatValue();

        System.out.println("Unboxed Float: " + unboxedFloat);

        byte byteValue = 10;

        Byte byteObject = Byte.valueOf(byteValue);

        System.out.println("Byte Object: " + byteObject);

        byte unboxedByte = byteObject.byteValue();

        System.out.println("Unboxed Byte: " + unboxedByte);

        short shortValue = 100;

        Short shortObject = Short.valueOf(shortValue);

        System.out.println("Short Object: " + shortObject);

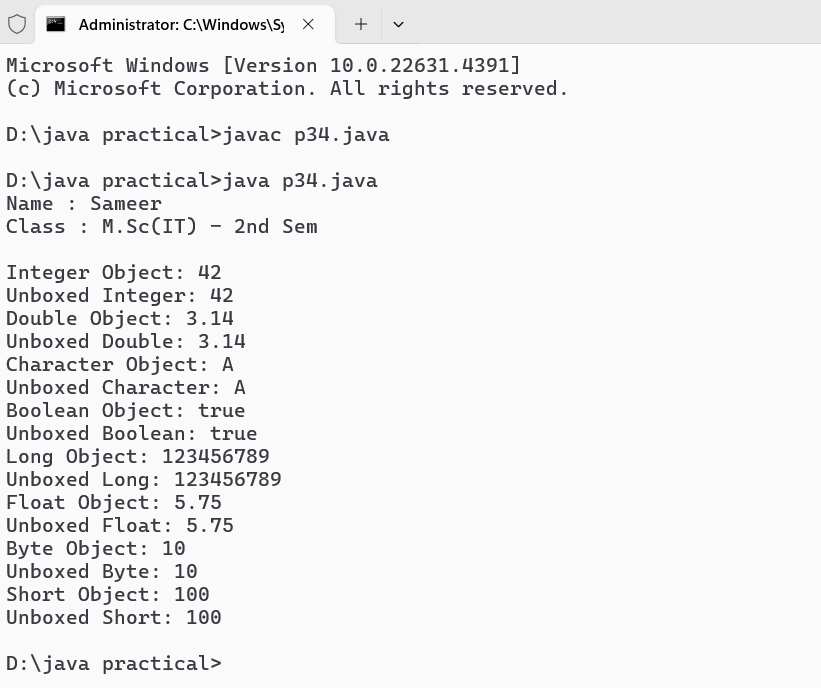
        short unboxedShort = shortObject.shortValue();

        System.out.println("Unboxed Short: " + unboxedShort);

    }

}

**Output :-**



**Program No. - 35**

**WAJP to demonstrate HashSet Class & its methods.**

import java.util.HashSet;

public class p35 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        HashSet<String> hashSet = new HashSet<>();

        hashSet.add("Apple");

        hashSet.add("Banana");

        hashSet.add("Orange");

        hashSet.add("Mango");

        hashSet.add("Grapes");

        System.out.println("HashSet: " + hashSet);

        if (hashSet.contains("Banana")) {

            System.out.println("HashSet contains Banana");

        } else {

            System.out.println("HashSet does not contain Banana");

        }

        hashSet.remove("Mango");

        System.out.println("HashSet after removing Mango: " + hashSet);

        System.out.println("Size of HashSet: " + hashSet.size());

        System.out.println("Iterating over HashSet:");

        for (String fruit : hashSet) {

            System.out.println(fruit);

        }

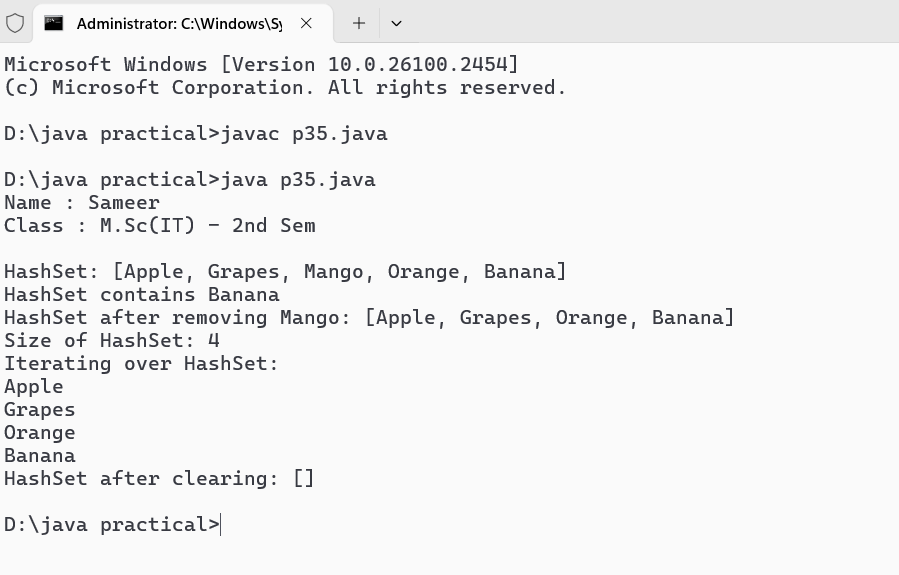
        hashSet.clear();

        System.out.println("HashSet after clearing: " + hashSet);

    }

}

**Output :-**



**Program No. - 36**

**WAJP to demonstrate ArrayList Class & its methods.**

import java.util.ArrayList;

public class p36 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        ArrayList<String> arrayList = new ArrayList<>();

        arrayList.add("Apple");

        arrayList.add("Banana");

        arrayList.add("Orange");

        arrayList.add("Mango");

        arrayList.add("Grapes");

        System.out.println("ArrayList: " + arrayList);

        String firstElement = arrayList.get(0);

        System.out.println("First element: " + firstElement);

        arrayList.set(2, "Pineapple");

        System.out.println("ArrayList after modifying element at index 2: " + arrayList);

        arrayList.remove("Banana");

        System.out.println("ArrayList after removing Banana: " + arrayList);

        System.out.println("Size of ArrayList: " + arrayList.size());

        if (arrayList.contains("Mango")) {

            System.out.println("ArrayList contains Mango");

        } else {

            System.out.println("ArrayList does not contain Mango");

        }

        System.out.println("Iterating over ArrayList:");

        for (String fruit : arrayList) {

            System.out.println(fruit);

        }

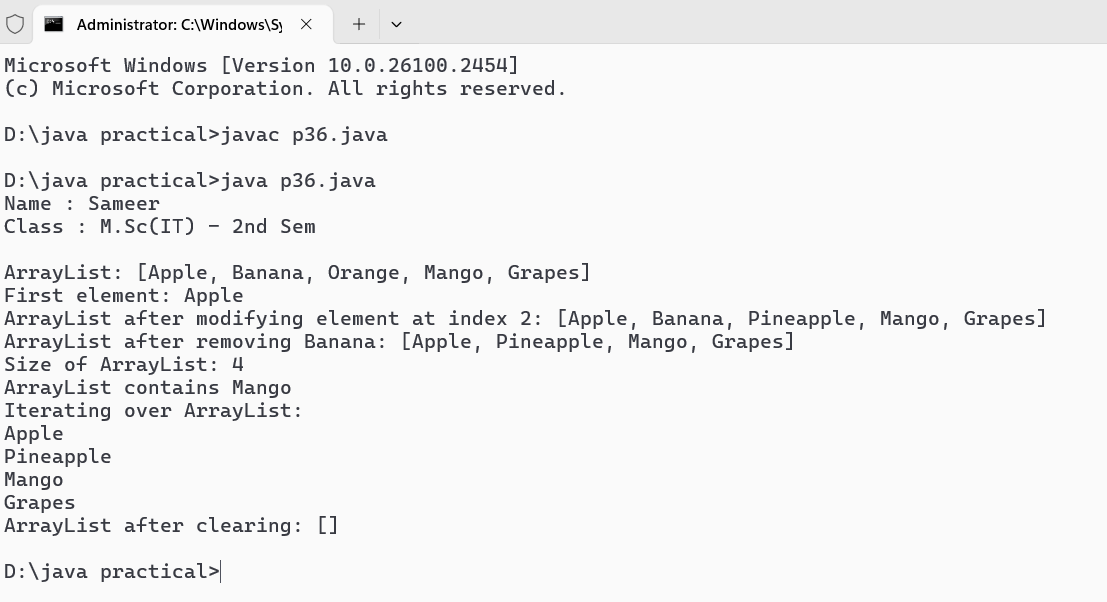
        arrayList.clear();

        System.out.println("ArrayList after clearing: " + arrayList);

    }

}

**Output :-**



**Program No. - 37**

**WAJP to copy a File.**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class p37 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Path sourcePath = Paths.get("source.txt");

        Path destinationPath = Paths.get("destination.txt");

        try {

            Files.copy(sourcePath, destinationPath);

            System.out.println("File copied successfully from " + sourcePath + " to " + destinationPath);

        } catch (IOException e) {

            System.err.println("Error occurred while copying the file: " + e.getMessage());

        }

    }

}

**Output :-**

**A screenshot of a computer

Description automatically generated**

**Program No. - 38**

**WAJP to Count the numbers of Characters in a File.**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class p38 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Path filePath = Paths.get("example.txt"); // Change this to your file path

        try {

            long characterCount = Files.lines(filePath).mapToLong(String::length).sum();

            System.out.println("Total number of characters in the file: " + characterCount);

        } catch (IOException e) {

            System.err.println("Error occurred while reading the file: " + e.getMessage());

        }

    }

}

**Output :-**

**A screenshot of a computer program

Description automatically generated**

**Program No. - 39**

**WAJP to demonstrate Object Serialization.**

import java.io.\*;

class Employee implements Serializable {

    private static final long serialVersionUID = 1L;

    private String name;

    private int age;

    public Employee(String name, int age) {

        this.name = name;

        this.age = age;

    }

    @Override

    public String toString() {

        return "Employee{name='" + name + "', age=" + age + '}';

    }

}

public class p39 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        Employee employee = new Employee("sameer kurre", 30);

        try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("employee.ser"))) {

            oos.writeObject(employee);

            System.out.println("Employee object serialized: " + employee);

        } catch (IOException e) {

            System.err.println("Error during serialization: " + e.getMessage());

        }

        try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("employee.ser"))) {

            Employee deserializedEmployee = (Employee) ois.readObject();

            System.out.println("Employee object deserialized: " + deserializedEmployee);

        } catch (IOException | ClassNotFoundException e) {

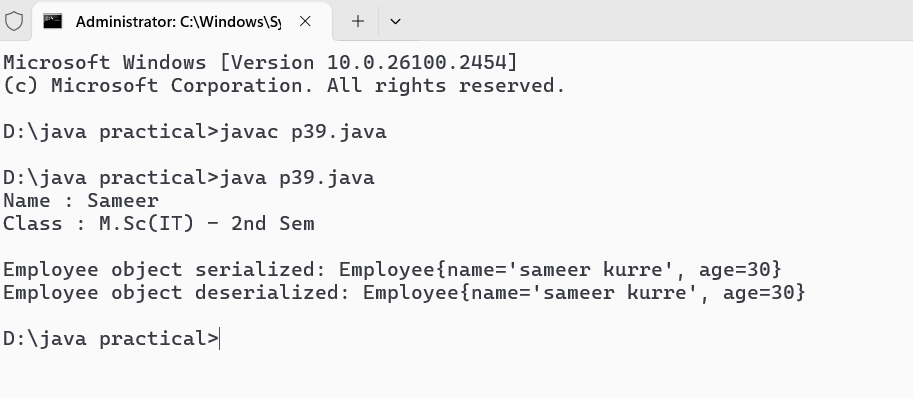
            System.err.println("Error during deserialization: " + e.getMessage());

        }

    }

}

**Output :-**



Program No. - 40.

WAJP to demonstrate Keyboard Event.

import java.awt.event.\*;

import javax.swing.\*;

public class p40 extends JFrame implements KeyListener{

JLabel jlabel;

JTextArea jtextarea;

public p40(){

jlabel = new JLabel();

jlabel.setText("Press a key......");

jlabel.setBounds(20, 50, 300, 20);

jtextarea = new JTextArea();

jtextarea.setBounds(20, 80, 300, 300);

jtextarea.addKeyListener(this);

add(jlabel);

add(jtextarea);

setTitle("KeyListener Example ");

setSize(400, 400);

setLayout(null);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

@Override

public void keyPressed(KeyEvent e){

jlabel.setText("Key Pressed : " + e.getKeyChar());

}

@Override

public void keyReleased(KeyEvent e){

jlabel.setText("Key Pressed : " + e.getKeyChar());

}

@Override

public void keyTyped(KeyEvent e){

jlabel.setText("Key Typed : " + e.getKeyChar());

}

public static void main(String[] args){

System.out.println("Name : Sameer");

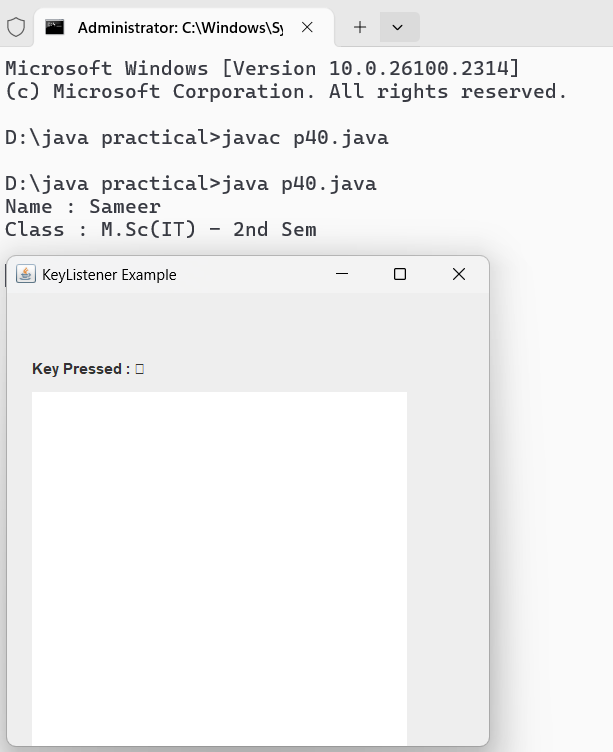
System.out.println("Class : M.Sc(IT) - 3rdSem\n");

p40 p = new p40();

}

}

**Output :-**



A screenshot of a computer

Description automatically generated

**Program No. - 41**

WAJP to demonstrate Mouse Event.

import java.awt.event.\*;

import javax.swing.\*;

public class p41 extends JFrame implements MouseListener {

    JLabel label;

    p41() {

        // Set up the JFrame

        setTitle("Mouse Event Example");

        setSize(300, 300);

        setLayout(null);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        // Create and set up the JLabel

        label = new JLabel();

        label.setBounds(20, 50, 200, 20);

        add(label);

        // Add mouse listener

        addMouseListener(this);

        // Make the frame visible

        setVisible(true);

    }

    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");

    }

    public void mousePressed(MouseEvent e) {

        label.setText("Mouse Pressed");

    }

    public void mouseReleased(MouseEvent e) {

        label.setText("Mouse Released");

    }

    public static void main(String[] args) {

System.out.println("Name : Sameer");

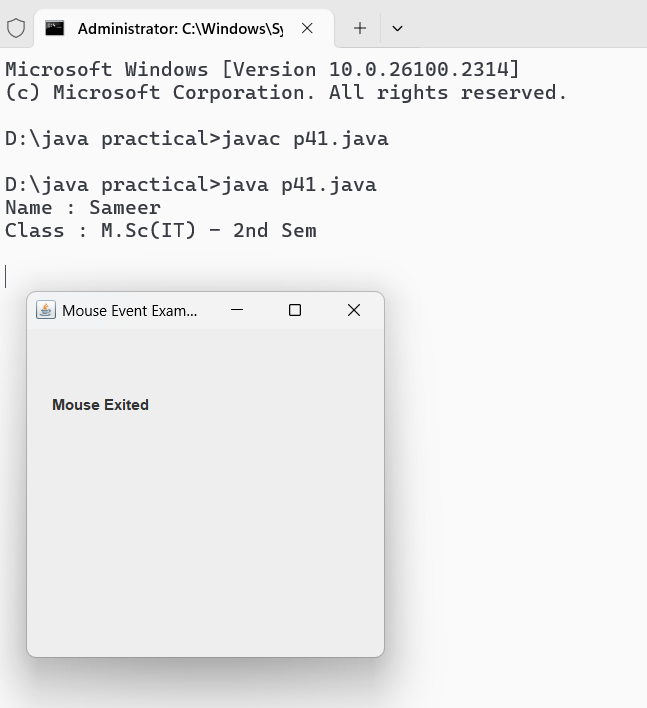
System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        p41 p = new p41();

    }

}

**Output :-**



A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Program No. - 42.

WAP to establish connection to the database.

import java.sql.\*;

public class p42 {

    public static void main(String[] args){

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        try{

            Class.forName("oracle.jdbc.driver.OracleDriver");

            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sameer", "sameer");

            System.out.println("Table Created Successfully");

            con.close();

        }

        catch(Exception e){

            System.out.println("Error : " + e);

        }

    }

}

**Output :-**

A screenshot of a computer program

Description automatically generated

Program No. - 43.

WAP to create a table named employee with feilds as emp\_id, emp\_name, age , dept.

import java.sql.\*;

public class p43 {

public static void main(String[] args){

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sameer", "sameer");

Statement stmt = con.createStatement();

stmt.executeQuery("Create table employee(emp\_id varchar(20), emp\_name varchar(20), age number, dept varchar(20))");

System.out.println("Table Created Successfully");

con.close();

}

catch(Exception e){

System.out.println("Error : " + e);

}

}

}

**Output :-**

**A screenshot of a computer program

Description automatically generated**

Program No. - 44.

WAP to create a table and drop it.

import java.sql.\*;

public class p44 {

public static void main(String[] args){

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sameer", "sameer");

Statement stmt = con.createStatement();

stmt.executeQuery("create table emp(id number, name varchar(20), salary number)");

System.out.println("Table Created successfully");

stmt.executeQuery("drop table emp");

System.out.println("Table Droped Successfully");

con.close();

}

catch(Exception e){

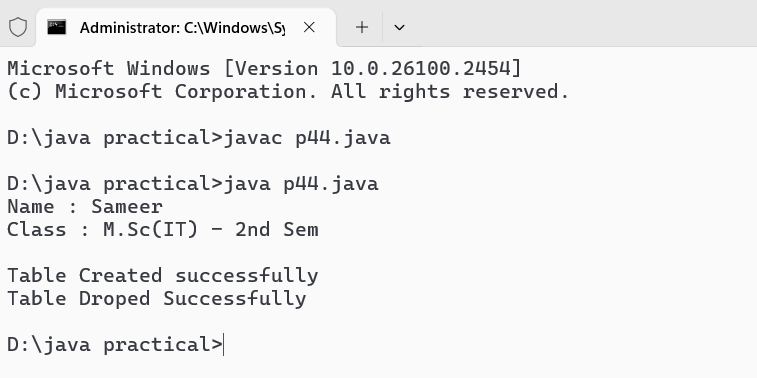
System.out.println("Error : " + e);

}

}

}

**Output :-**

****

Program No. - 45.

WAP to insert multiple rows in a table using prepared statement.

import java.sql.\*;

public class p45 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

Connection con = null;

Statement stmt = null;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sameer", "sameer");

stmt = con.createStatement();

// Use single quotes for SQL string values

stmt.executeUpdate("insert into emp values(101, 'Sameer', 100000)");

stmt.executeUpdate("insert into emp values(102, 'Manish', 70000)");

stmt.executeUpdate("insert into emp values(103, 'Ram', 60000)");

stmt.executeUpdate("insert into emp values(104, 'Sneha', 110000)");

stmt.executeUpdate("insert into emp values(105, 'Esha', 70000)");

System.out.println("Data Inserted successfully");

} catch (Exception e) {

System.out.println("Error : " + e);

} finally {

// Close resources

try {

if (stmt != null) stmt.close();

if (con != null) con.close();

} catch (SQLException e) {

System.out.println("Error closing resources: " + e);

}

}

}

}

**Output :-**

**A screenshot of a computer program

Description automatically generated**

Program No. - 46.

WAP to display contents of a table on the console.

import java.sql.\*;

public class p46 {

public static void main(String[] args){

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sameer", "sameer");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from emp");

System.out.println("Table Contents : ");

while(rs.next()){

System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3));

}

con.close();

}

catch(Exception e){

System.out.println("Error : " + e);

}

}

}

**Output :-**

**A screenshot of a computer

Description automatically generated**

Program No. - 47.

WAP to update rows using result set.

import java.sql.\*;

public class p46 {

public static void main(String[] args){

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

try{

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "sameer", "sameer");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from emp");

System.out.println("Table Contents : ");

while(rs.next()){

System.out.println(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3));

}

stmt.executeQuery("update emp set name = 'mamta' where id = 101");

System.out.println("Updates Table Contents : ");

ResultSet rs1 = stmt.executeQuery("select \* from emp");

while (rs1.next()) {

System.out.println(rs1.getInt(1) + " " + rs1.getString(2) + " " + rs1.getInt(3));

}

con.close();

}

catch(Exception e){

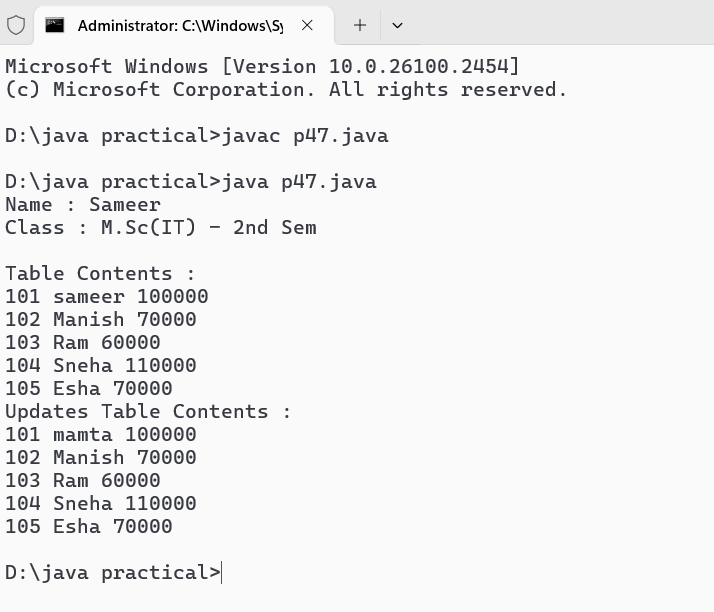
System.out.println("Error : " + e);

}

}

}

**Output :-**

****

Program No. - 48.

WAP to describe the functions of metadata objects.(resultset & database)

Program No. - 49.

WAP to demonstrate the ArrayList class.

import java.util.ArrayList;

public class p49 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

ArrayList<String> fruits = new ArrayList<>();

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Cherry");

fruits.add("Date");

System.out.println("Fruits in the ArrayList: " + fruits);

String firstFruit = fruits.get(0);

System.out.println("First fruit: " + firstFruit);

fruits.remove("Banana");

System.out.println("After removing Banana: " + fruits);

int size = fruits.size();

System.out.println("Size of the ArrayList: " + size);

System.out.println("Iterating through the ArrayList:");

for (String fruit : fruits) {

System.out.println(fruit);

}

}

}

**Output :-**

A screenshot of a computer program

Description automatically generated

Program No. - 50.

WAP to demonstrate the HashSet class.

import java.util.HashSet;

public class p50 {

public static void main(String[] args) {

System.out.println("Name : Sameer");

System.out.println("Class : M.Sc(IT) - 3rdSem\n");

HashSet<String> fruits = new HashSet<>();

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Cherry");

fruits.add("Date");

fruits.add("Banana");

System.out.println("Fruits in the HashSet: " + fruits);

if (fruits.contains("Apple")) {

System.out.println("Apple is present in the HashSet.");

} else {

System.out.println("Apple is not present in the HashSet.");

}

fruits.remove("Cherry");

System.out.println("After removing Cherry: " + fruits);

int size = fruits.size();

System.out.println("Size of the HashSet: " + size);

System.out.println("Iterating through the HashSet:");

for (String fruit : fruits) {

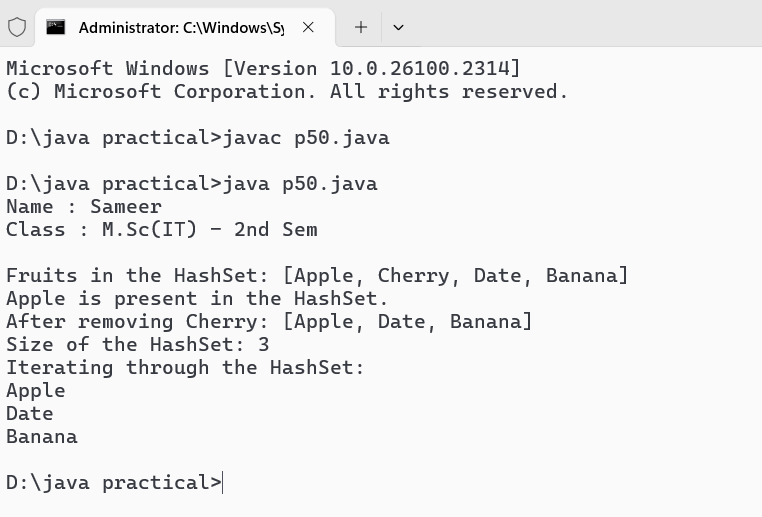
System.out.println(fruit);

}

}

}

Output :-



Program No. - 51.

WAP to demonstrate the HashMap class.

import java.util.HashMap;

import java.util.Map;

public class p51 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        HashMap<String, Integer> map = new HashMap<>();

        map.put("sameer", 30);

        map.put("manish", 25);

        map.put("tuleshwar", 35);

        map.put("dhananjay", 28);

        System.out.println("Initial HashMap: " + map);

        String keyToRetrieve = "sameer";

        Integer value = map.get(keyToRetrieve);

        System.out.println("Value associated with key '" + keyToRetrieve + "': " + value);

        String keyToCheck = "tuleshwar";

        if (map.containsKey(keyToCheck)) {

            System.out.println("Key '" + keyToCheck + "' exists in the HashMap.");

        } else {

            System.out.println("Key '" + keyToCheck + "' does not exist in the HashMap.");

        }

        System.out.println("\nIterating through the HashMap:");

        for (Map.Entry<String, Integer> entry : map.entrySet()) {

            System.out.println("Key: " + entry.getKey() + ", Value: " + entry.getValue());

        }

        String keyToRemove = "manish";

        map.remove(keyToRemove);

        System.out.println("\nHashMap after removing key '" + keyToRemove + "': " + map);

        System.out.println("Size of the HashMap: " + map.size());

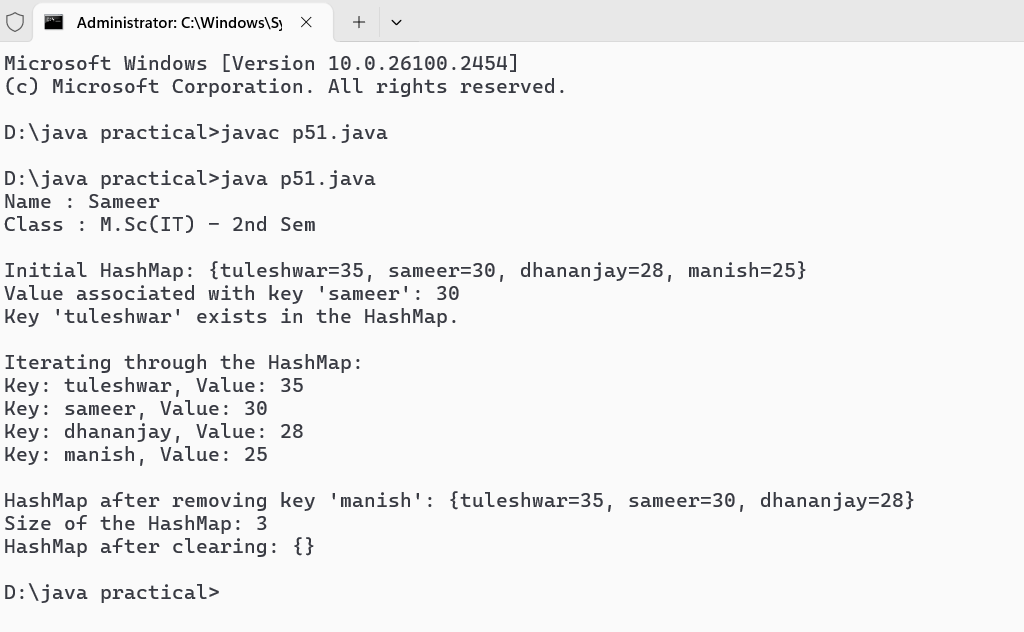
        map.clear();

        System.out.println("HashMap after clearing: " + map);

    }

}

**Output :-**

****

Program No. - 52.

WAP to demonstrate the Vector class.

public class p52 {

    public double x;

    public double y;

    public p52(double x, double y) {

        this.x = x;

        this.y = y;

    }

    @Override

    public String toString() {

        return "p52(" + x + ", " + y + ")";

    }

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        p52 v1 = new p52(2, 3);

        p52 v2 = new p52(4, 5);

        System.out.println("p52 1: " + v1);

        System.out.println("p52 2: " + v2);

        p52 v3 = new p52(v1.x + v2.x, v1.y + v2.y);

        System.out.println("v1 + v2 = " + v3);

        p52 v4 = new p52(v1.x - v2.x, v1.y - v2.y);

        System.out.println("v1 - v2 = " + v4);

        double scalar = 3;

        p52 v5 = new p52(v1.x \* scalar, v1.y \* scalar);

        System.out.println("v1 \* " + scalar + " = " + v5);

        double dotProduct = (v1.x \* v2.x) + (v1.y \* v2.y);

        System.out.println("Dot product of v1 and v2 = " + dotProduct);

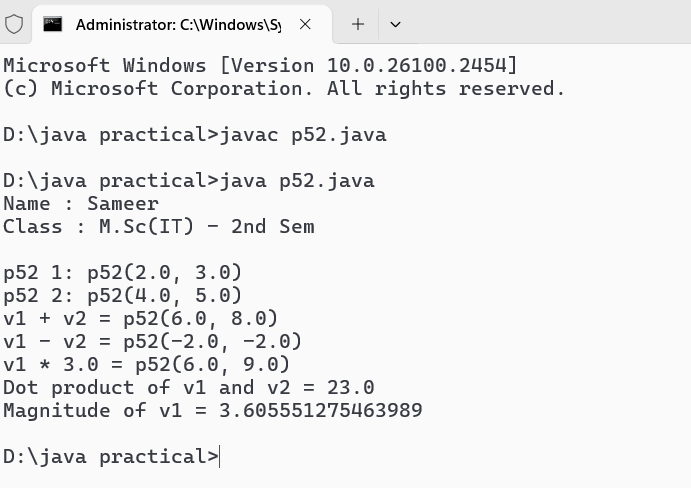
        double magnitudeV1 = Math.sqrt(v1.x \* v1.x + v1.y \* v1.y);

        System.out.println("Magnitude of v1 = " + magnitudeV1);

    }

}

**Output :-**



Program No. - 53.

WAP to demonstrate the LinkedList class.

import java.util.LinkedList;

public class p53 {

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        LinkedList<String> linkedList = new LinkedList<>();

        linkedList.add("Apple");

        linkedList.add("Banana");

        linkedList.add("Cherry");

        linkedList.addFirst("Mango");

        linkedList.addLast("Grapes");

        System.out.println("LinkedList: " + linkedList);

        linkedList.remove("Banana");

        linkedList.remove(2);

        linkedList.removeFirst();

        linkedList.removeLast();

        System.out.println("After removals: " + linkedList);

        String firstElement = linkedList.getFirst();

        String lastElement = linkedList.getLast();

        System.out.println("First Element: " + firstElement);

        System.out.println("Last Element: " + lastElement);

        System.out.println("Size of LinkedList: " + linkedList.size());

        System.out.println("Is LinkedList empty? " + linkedList.isEmpty());

        System.out.println("Does LinkedList contain 'Apple'? " + linkedList.contains("Apple"));

        String[] array = linkedList.toArray(new String[0]);

        System.out.println("Array from LinkedList: ");

        for (String s : array) {

            System.out.println(s);

        }

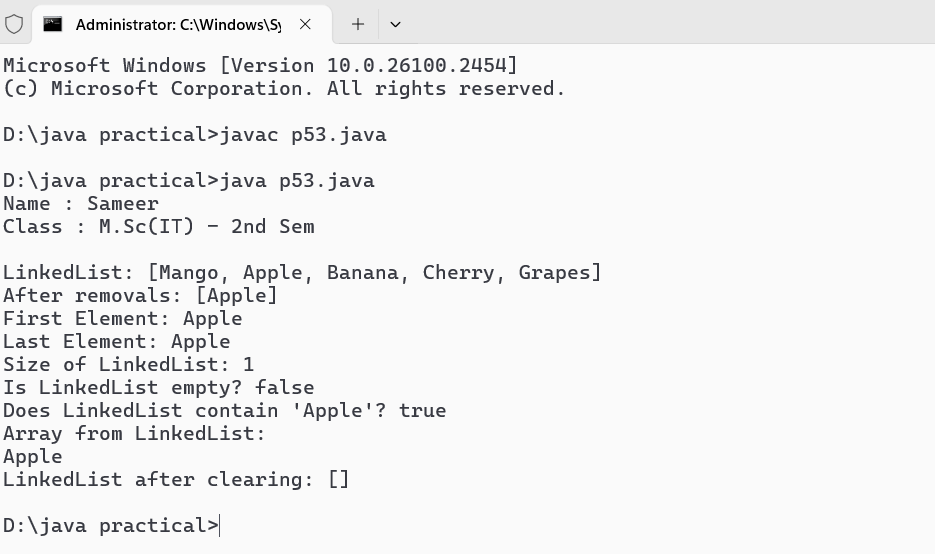
        linkedList.clear();

        System.out.println("LinkedList after clearing: " + linkedList);

    }

}

**Output :-**



Program No. - 54.

WAP to demonstrate the JTextField class.

import javax.swing.\*;

class p54

{

    public static void main(String args[])

    {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        JFrame f= new JFrame("TextField Example");

        JTextField t1;

        t1 = new JTextField("Welcome to Raipur.");

        t1.setBounds(50,100, 200,30);

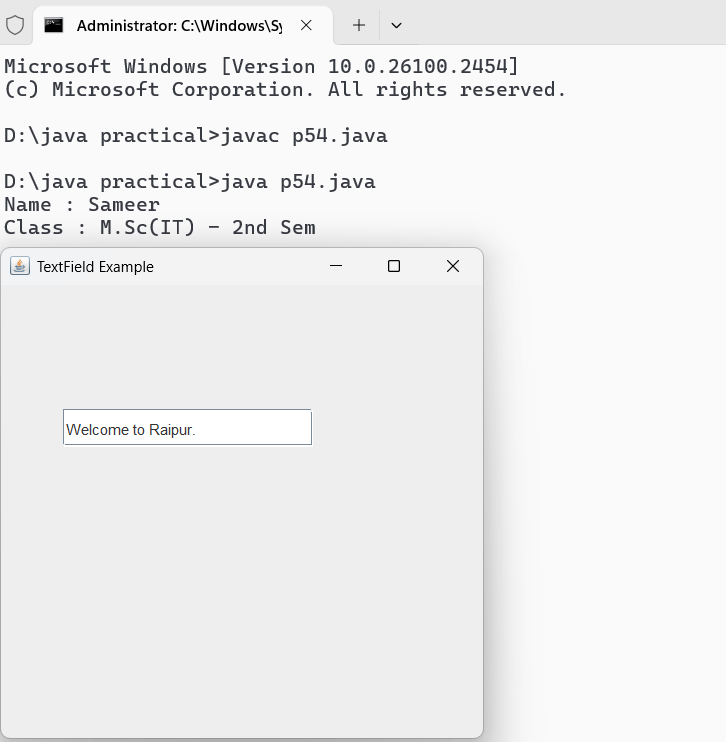
        f.add(t1);

        f.setSize(400,400);

        f.setLayout(null);

        f.setVisible(true);

    }

}      
  
Output :-   


Program No. - 55.

WAP to demonstrate the JButton class.

import javax.swing.\*;

class p55

{

    public static void main(String args[])

    {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        JFrame jframe = new JFrame();

        jframe.setTitle("Button Example");

        jframe.setLayout(null);

        jframe.setSize(700, 700);

        jframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        JButton jbutton = new JButton();

        jbutton.setText("I am a button.");

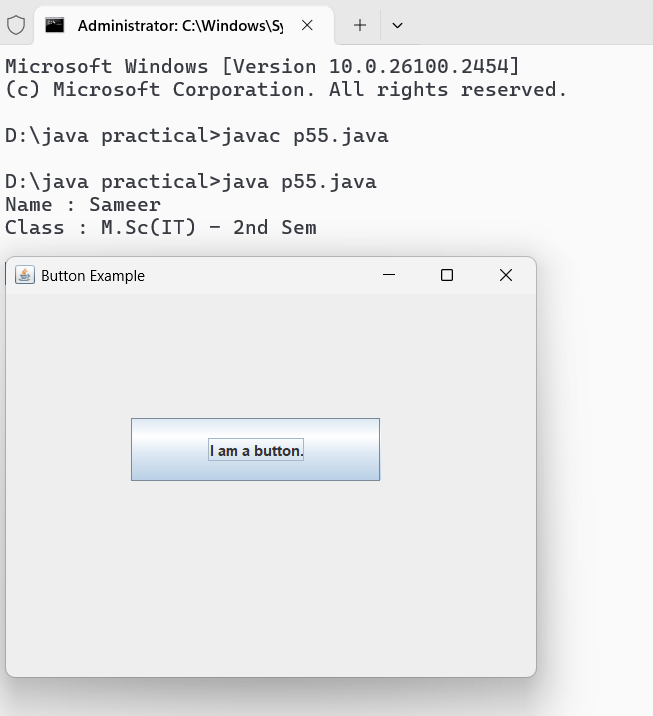
        jbutton.setBounds(100, 100, 200, 50);

        jframe.add(jbutton);

        jframe.setVisible(true);

    }

}

**Output :-   
**

Program No. - 56.

WAP to demonstrate the JToggleButton class.

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class p56 extends JFrame {

    private JToggleButton toggleButton;

    private JLabel label;

    public p56() {

        setTitle("JToggleButton Demo");

        setSize(300, 200);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(new FlowLayout());

        toggleButton = new JToggleButton("Off");

        toggleButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                if (toggleButton.isSelected()) {

                    toggleButton.setText("On");

                    label.setText("Button is ON");

                } else {

                    toggleButton.setText("Off");

                    label.setText("Button is OFF");

                }

            }

        });

        label = new JLabel("Button is OFF");

        add(toggleButton);

        add(label);

    }

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        SwingUtilities.invokeLater(new Runnable() {

            @Override

            public void run() {

                p56 demo = new p56();

                demo.setVisible(true);

            }

        });

    }

}

**Output :-**

A screenshot of a computer

Description automatically generated

Program No. - 57.

WAP to demonstrate the JCheckbox class.

import java.awt.\*;

import javax.swing.\*;

public class p57 extends JFrame {

    public p57() {

        setTitle("JCheckBox Demo");

        setSize(200, 100);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(new FlowLayout());

        JCheckBox checkBox1 = new JCheckBox("Option 1");

        JCheckBox checkBox2 = new JCheckBox("Option 2");

        JButton button = new JButton("Get Selection");

        button.addActionListener(e -> {

            String message = "Selected options: ";

            if (checkBox1.isSelected()) {

                message += "Option 1 ";

            }

            if (checkBox2.isSelected()) {

                message += "Option 2 ";

            }

            if (!checkBox1.isSelected() && !checkBox2.isSelected()) {

                message += "None";

            }

            JOptionPane.showMessageDialog(this, message);

        });

        add(checkBox1);

        add(checkBox2);

        add(button);

    }

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        SwingUtilities.invokeLater(() -> {

            p57 demo = new p57();

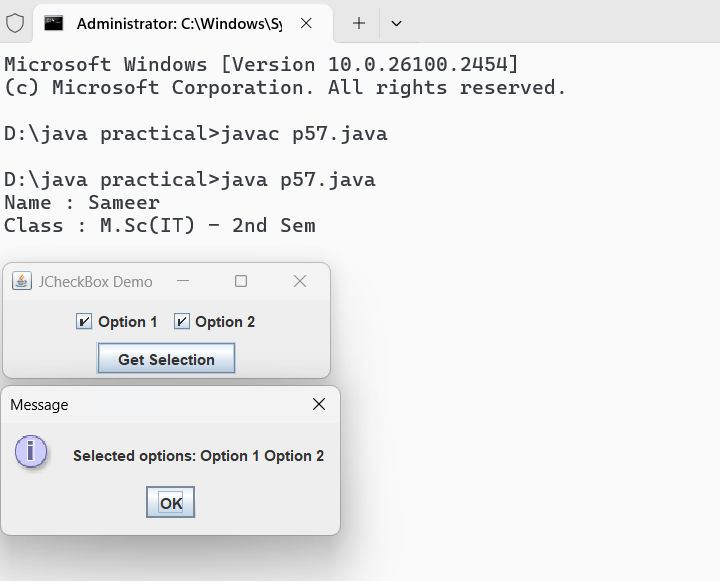
            demo.setVisible(true);

        });

    }

}

**Output :-**



Program No. - 58.

WAP to demonstrate the JRadioButton class.

import java.awt.\*;

import javax.swing.\*;

public class p58 extends JFrame {

    public p58() {

        setTitle("JRadioButton Demo");

        setSize(300, 150);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(new FlowLayout());

        JRadioButton option1 = new JRadioButton("Option 1");

        JRadioButton option2 = new JRadioButton("Option 2");

        ButtonGroup group = new ButtonGroup();

        group.add(option1);

        group.add(option2);

        JButton button = new JButton("Get Selection");

        button.addActionListener(e -> {

            String message = "Selected option: ";

            if (option1.isSelected()) {

                message += "Option 1";

            } else if (option2.isSelected()) {

                message += "Option 2";

            } else {

                message += "None";

            }

            JOptionPane.showMessageDialog(this, message);

        });

        add(option1);

        add(option2);

        add(button);

    }

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        SwingUtilities.invokeLater(() -> {

            p58 demo = new p58();

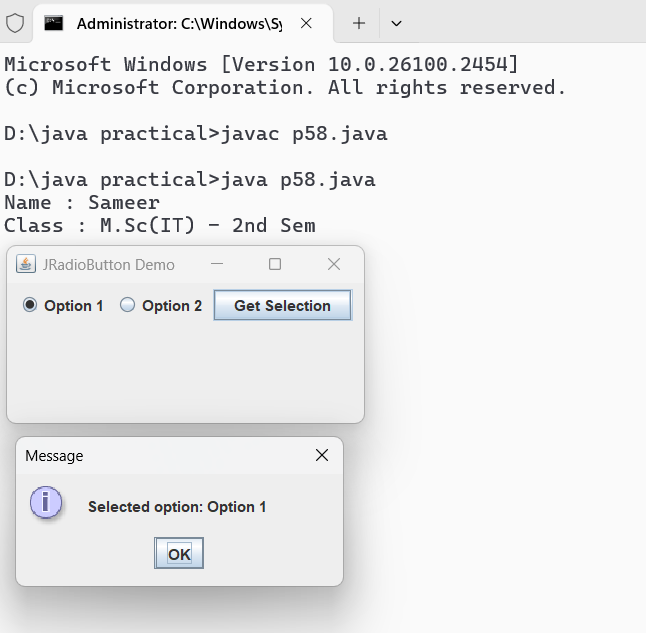
            demo.setVisible(true);

        });

    }

}

**Output :-**

****

Program No. - 59.

WAP to demonstrate the JComboBox class.

import java.awt.\*;

import javax.swing.\*;

public class p59 extends JFrame {

    public p59() {

        setTitle("JComboBox Demo");

        setSize(300, 150);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(new FlowLayout());

        String[] items = {"Option 1", "Option 2", "Option 3"};

        JComboBox<String> comboBox = new JComboBox<>(items);

        JButton button = new JButton("Get Selection");

        button.addActionListener(e -> {

            String selectedItem = (String) comboBox.getSelectedItem();

            JOptionPane.showMessageDialog(this, "Selected option: " + selectedItem);

        });

        add(comboBox);

        add(button);

    }

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        SwingUtilities.invokeLater(() -> {

            p59 demo = new p59();

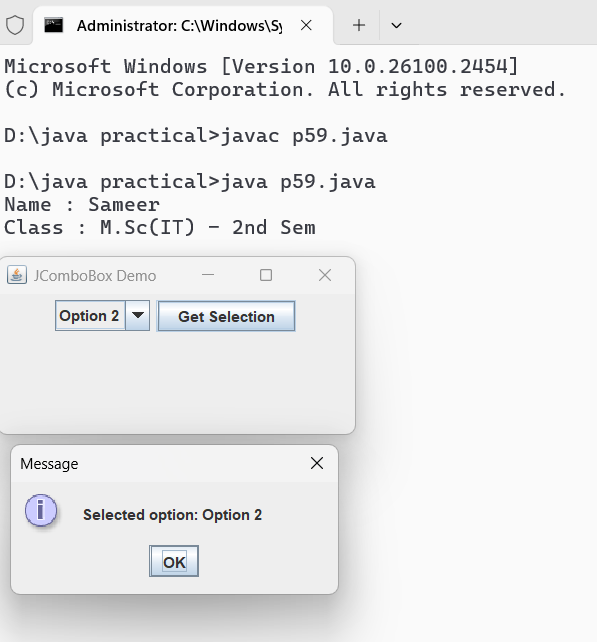
            demo.setVisible(true);

        });

    }

}

**Output :-**

****

Program No. - 60.

WAP to demonstrate the JList class

import javax.swing.\*;

import java.awt.\*;

public class p60 extends JFrame {

    public p60() {

        setTitle("JList Demo");

        setSize(300, 200);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setLayout(new FlowLayout());

        // Sample data for the list

        String[] items = {"Item 1", "Item 2", "Item 3", "Item 4", "Item 5"};

        JList<String> list = new JList<>(items);

        list.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

        JScrollPane scrollPane = new JScrollPane(list);

        scrollPane.setPreferredSize(new Dimension(150, 100));

        JButton button = new JButton("Get Selection");

        button.addActionListener(e -> {

            String selectedItem = list.getSelectedValue();

            JOptionPane.showMessageDialog(this, "Selected item: " + (selectedItem != null ? selectedItem : "None"));

        });

        add(scrollPane);

        add(button);

    }

    public static void main(String[] args) {

        System.out.println("Name : Sameer");

        System.out.println("Class : M.Sc(IT) - 3rdSem\n");

        SwingUtilities.invokeLater(() -> {

            p60 demo = new p60();

            demo.setVisible(true);

        });

    }

}

Output :-

