# Name: Shaikh Faiz Ahmed

# Subject: Core Java Practical

# 1.Java basics

* 1. Write a Java program to swap two variables.

**CODE:**

public class SwapVariables {

public static void main(String[] args) { int x = 5;

int y = 10;

/ Display the values before swapping System.out.println("Before swapping:"); System.out.println("x = " + x); System.out.println("y = " + y);

/ Swap the values using a temporary variable int temp = x;

x = y;

y = temp;

/ Display the values after swapping System.out.println("After swapping:"); System.out.println("x = " + x); System.out.println("y = " + y);

}

}

#### Output:

* 1. Write a Java program to print the area of a circle.

#### CODE:

import java.util.Scanner; public class CircleArea {

public static void main(String[] args) { double radius, area;

/ create an instance of Scanner class to read input from console Scanner scanner = new Scanner(System.in);

/ read radius of the circle from console System.out.print("Enter the radius of the circle: "); radius = scanner.nextDouble();

/ calculate the area of the circle using the formula A = pi \* r \* r area = Math.PI \* radius \* radius;

/ display the result

System.out.println("The area of the circle is: " + area);

}

}

#### Output:

* 1. Write a Java program to print the sum (addition), multiply, subtract, divide and remainder of two numbers.

#### Code:

import java.util.Scanner;

public class ArithmeticOperations { public static void main(String[] args) {

double num1, num2, sum, product, difference, quotient, remainder;

/ create an instance of Scanner class to read input from console Scanner scanner = new Scanner(System.in);

/ read the two numbers from console System.out.print("Enter the ﬁrst number: "); num1 = scanner.nextDouble(); System.out.print("Enter the second number: "); num2 = scanner.nextDouble();

/ perform arithmetic operations sum = num1 + num2;

product = num1 \* num2; difference = num1 - num2; quotient = num1 / num2; remainder = num1 % num2;

/ display the results System.out.println("Sum: " + sum); System.out.println("Product: " + product); System.out.println("Difference: " + difference); System.out.println("Quotient: " + quotient);

System.out.println("Remainder: " + remainder);

}

}

#### Output:

* 1. Write a Java program that takes three numbers as input to calculate and print the average of the numbers.

#### Code:

import java.util.Scanner;

public class AverageOfThreeNumbers { public static void main(String[] args) {

double num1, num2, num3, average;

/ create an instance of Scanner class to read input from console Scanner scanner = new Scanner(System.in);

/ read the three numbers from console System.out.print("Enter the ﬁrst number: "); num1 = scanner.nextDouble(); System.out.print("Enter the second number: "); num2 = scanner.nextDouble(); System.out.print("Enter the third number: "); num3 = scanner.nextDouble();

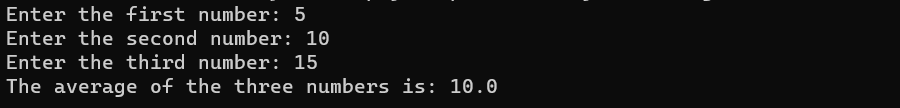
/ calculate the average of the three numbers average = (num1 + num2 + num3) / 3;

/ display the result

System.out.println("The average of the three numbers is: " + average);

}

}

**Output:**

# USE OF OPERATORS

* 1. WRITE A JAVA PROGRAM TO CHECK NUMBER IS DIVISIBLE BY 7 OR NOT.

#### CODE:

import java.util.Scanner; public class DivisibleBySeven {

public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.print("Enter a number: "); int num = sc.nextInt();

if(num % 7 == 0) {

System.out.println(num + " is divisible by 7");

} else {

System.out.println(num + " is not divisible by 7");

}

sc.close();

}

}

OUTPUT:

* 1. WRITE A PROGRAM TO FIND MAXIMUM FROM 3 NUMBERS CODE:

public class MaxOfThreeNumbers {

public static void main(String[] args) {

int num1 = 10, num2 = 20, num3 = 15;

int max = num1;

if (num2 > max) {

max = num2;

}

if (num3 > max) {

max = num3;

}

System.out.println("Maximum of three numbers is: " + max);

}

} OUTPUT:

* 1. WRITE A JAVA PROGRAM TO READ DAY CODE AND DISPLAY DAY NAME USING SWITCH CASE

#### CODE:

import java.util.Scanner; public class DayName {

public static void main(String[] args) {

/ create a Scanner object to read input from user

Scanner scanner = new Scanner(System.in);

/ read day code from user

System.out.print("Enter day code (1-7): ");

int dayCode = scanner.nextInt();

/ use switch case to determine day name

String dayName = "";

switch(dayCode) {

case 1:

dayName = "Sunday";

break;

case 2:

dayName = "Monday";

break;

case 3:

dayName = "Tuesday";

break;

case 4:

dayName = "Wednesday";

break;

case 5:

dayName = "Thursday";

break;

case 6:

dayName = "Friday";

break;

case 7:

dayName = "Saturday";

break;

default:

System.out.println("Invalid day code entered!");

System.exit(0);

}

System.out.println("Day name: " + dayName);

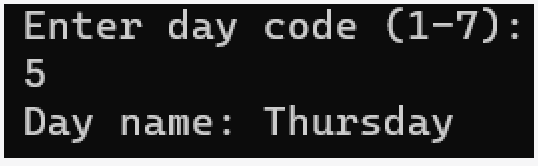
/ close the scanner object

scanner.close();

}

}

OUTPUT:



* 1. WRITE A JAVA PROGRAM THAT TAKES A NUMBER AS INPUT AND PRINTS MULTIPLICATION TABLE UPTO 10

#### CODE:

import java.util.Scanner;

public class MultiplicationTable {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

int number = input.nextInt();

for (int i = 1; i <= 10; i++) {

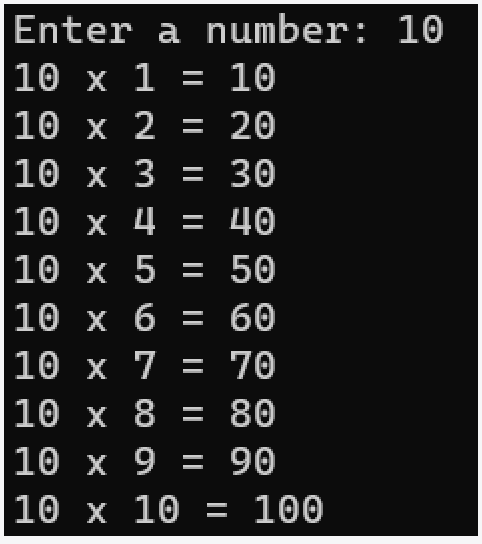
int result = number \* i;

System.out.println(number + " x " + i + " = " + result);

}

}

}

OUTPUT:

* 1. Write a Java program and compute the sum of the digits of an integer.

#### CODE:

import java.util.Scanner; public class SumOfDigits {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter an integer: ");

int num = input.nextInt();

int sum = 0;

while (num > 0) {

sum += num % 10;

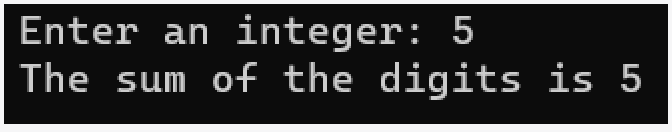
num /= 10;

}

System.out.println("The sum of the digits is " + sum);

}

}

**OUTPUT:**

# JAVA DATA TYPES

1. Write a Java program to ﬁnd sum of array elements of size 10.

#### CODE:

public class SumOfArrayElements {

public static void main(String[] args) {

/ Initialize an array of size 10

int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

/ Initialize a variable to store the sum

int sum = 0;

/ Loop through each element of the array and add it to the sum

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

sum += arr[i];

}

/ Print the sum

System.out.println("The sum of the elements of the array is: " + sum);

}

}

OUTPUT:



1. Designed a class SortData that contains the method asec() and desc() to sort array elements

**CODE:**

import java.util.Arrays; public class SortData {

public static void main(String[] args) {

int[] data = {3, 7, 1, 9, 2, 5, 8, 4, 6};

/ Sort data in ascending order

Arrays.sort(data);

System.out.println("Ascending order: " + Arrays.toString(data));

/ Sort data in descending order

for (int i = 0; i < data.length / 2; i++) {

int temp = data[i];

data[i] = data[data.length - 1 - i];

data[data.length - 1 - i] = temp;

}

System.out.println("Descending order: " + Arrays.toString(data));

}

}

#### OUTPUT:



1. Find the smallest and largest element from the array

#### CODE:

import java.util.Arrays;

public class SmallestAndLargestElement {

public static void main(String[] args) {

int[] arr = {3, 7, 1, 9, 4, 2, 8, 6, 5};

/ Find the smallest element

int min = arr[0];

for (int i = 1; i < arr.length; i++) {

if (arr[i] < min) {

min = arr[i];

}

}

System.out.println("Smallest element: " + min);

/ Find the largest element

int max = arr[0];

for (int i = 1; i < arr.length; i++) {

if (arr[i] > max) {

max = arr[i];

}

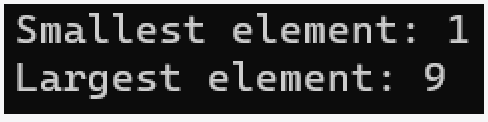
}

System.out.println("Largest element: " + max);

}

}

OUTPUT:



# Methods and Constructors

* 1. Designed a Box class to ﬁnd volume of a Box using class method.

### CODE:

class Box {

double length;

double width;

double height;

/ constructor

Box(double l, double w, double h) {

length = l;

width = w;

height = h;

}

/ class method to ﬁnd volume

static double volume(Box b) {

return b.length \* b.width \* b.height;

}

}

public class Main {

public static void main(String[] args) {

/ create a Box object

Box myBox = new Box(10, 5, 2);

/ call the class method to ﬁnd volume

double volume = Box.volume(myBox);

/ print the volume

System.out.println("The volume of the box is: " + volume);

}

}

**OUTPUT:**



* 1. Designed a Box class, initialize object using constructor and ﬁnd volume of a box

**CODE:**

class Box {

double length;

double width;

double height;

/ Constructor

Box(double l, double w, double h) {

length = l;

width = w;

height = h;

}

/ Method to calculate volume

double volume() {

return length \* width \* height;

}

}

public class Main {

public static void main(String[] args) {

/ Create a Box object with dimensions 5x3x2

Box myBox = new Box(5, 3, 2);

/ Calculate and print the volume of the box

System.out.println("Volume of the box is: " + myBox.volume());

}

}

**OUTPUT:**



* 1. Designed a Box class with multiple constructor to initialize class object and ﬁnd volume of a box.

###### CODE:

public class Box { private double length; private double width; private double height;

public Box() { this.length = 0;

this.width = 0;

this.height = 0;

}

public Box(double length, double width, double height) { this.length = length;

this.width = width; this.height = height;

}

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

this.width = width; this.height = 0;

}

public double getVolume() { return length \* width \* height;

}

public static void main(String[] args) { Box box1 = new Box();

System.out.println("Box 1 volume: " + box1.getVolume());

Box box2 = new Box(10, 20, 30);

System.out.println("Box 2 volume: " + box2.getVolume());

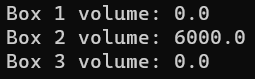
Box box3 = new Box(5, 10);

System.out.println("Box 3 volume: " + box3.getVolume());

}

}

**OUTPUT:**



# 5.Inheritance

1. Write a java program to implement single level inheritance.

**Code:**

class Animal { public void eat() {

System.out.println("The animal is eating.");

}

}

class Dog extends Animal { public void bark() {

System.out.println("The dog is barking.");

}

}

public class Main {

public static void main(String[] args) { Dog myDog = new Dog(); myDog.eat();

myDog.bark();

}

}

Output:



1. Write a java program to implement multilevel inheritance.

### Code:

class Animal { void eat() {

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

}

}

class Dog extends Animal { void bark() {

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

}

}

class GermanShepherd extends Dog { void play() {

System.out.println("German Shepherd is playing...");

}

}

public class MultilevelInheritanceExample { public static void main(String[] args) {

GermanShepherd gs = new GermanShepherd(); gs.eat(); // inherited from Animal class gs.bark(); // inherited from Dog class

gs.play(); // specific to GermanShepherd class

}

}

#### Output:

1. Write a java program to implement multiple inheritance.

#### Code:

interface A {

void methodA();

}

interface B {

void methodB();

}

class MyClass implements A, B { public void methodA() {

System.out.println("Method A");

}

public void methodB() { System.out.println("Method B");

}

}

public class MultipleInheritanceExample { public static void main(String[] args) { MyClass obj = new MyClass();

obj.methodA(); obj.methodB();

}

}

**Output:**



# Packages and Arrays

* 1. Create a package MyMath, Add the necessary classes and import the package in java class.

**Code:**

##### 1. Create a new folder named MyMath in your project directory

package MyMath;

public class Calculator {

public static int add(int a, int b) { return a + b;

}

public static int subtract(int a, int b) { return a - b;

}

public static int multiply(int a, int b) { return a \* b;

}

public static int divide(int a, int b) { return a / b;

}

}

**2. Save the Calculator.java ﬁle.**

**3**. **In your Java class where you want to use the Calculator class, add the following import statement at the top of your ﬁle:**

import MyMath.Calculator;

public class MyJavaClass {

public static void main(String[] args) { int a = 10;

int b = 5;

int sum = Calculator.add(a, b);

int difference = Calculator.subtract(a, b); int product = Calculator.multiply(a, b); int quotient = Calculator.divide(a, b);

System.out.println("The sum of " + a + " and " + b + " is " + sum); System.out.println("The difference between " + a + " and " + b + " is " +

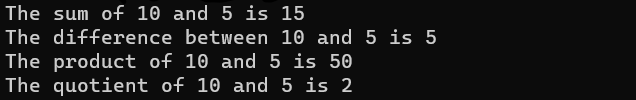
difference);

System.out.println("The product of " + a + " and " + b + " is " + product); System.out.println("The quotient of " + a + " and " + b + " is " + quotient);

}

}

#### Output:



* 1. Write a java program to add two matrices and print the resultant matrix.

#### Code:

import java.util.Scanner; public class MatrixAddition {

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

System.out.print("Enter the number of rows: "); int rows = scanner.nextInt();

System.out.print("Enter the number of columns: "); int cols = scanner.nextInt();

int[][] matrix1 = new int[rows][cols]; int[][] matrix2 = new int[rows][cols];

System.out.println("Enter the elements of matrix1:"); for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) { matrix1[i][j] = scanner.nextInt();

}

}

System.out.println("Enter the elements of matrix2:"); for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) { matrix2[i][j] = scanner.nextInt();

}

}

int[][] result = new int[rows][cols]; for (int i = 0; i < rows; i++) {

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

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

}

}

System.out.println("The resultant matrix is:"); for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) { System.out.print(result[i][j] + " ");

}

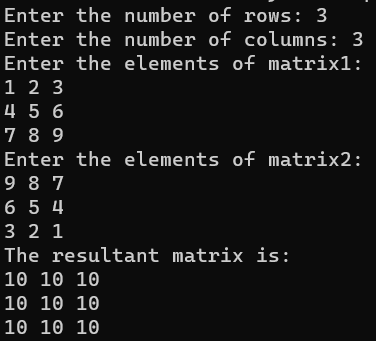
System.out.println();

}

}

}

#### Output:



* 1. Write a java program for multiplying two matrices and print the product for the same.

#### Code:

import java.util.Scanner;

public class MatrixMultiplication { public static void main(String[] args) {

Scanner scanner = new Scanner(System.in); System.out.print("Enter the number of rows for matrix1: "); int rows1 = scanner.nextInt();

System.out.print("Enter the number of columns for matrix1: "); int cols1 = scanner.nextInt();

System.out.print("Enter the number of rows for matrix2: "); int rows2 = scanner.nextInt();

System.out.print("Enter the number of columns for matrix2: "); int cols2 = scanner.nextInt();

if (cols1 != rows2) {

System.out.println("The matrices cannot be multiplied."); return;

}

int[][] matrix1 = new int[rows1][cols1]; int[][] matrix2 = new int[rows2][cols2];

System.out.println("Enter the elements of matrix1:"); for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols1; j++) { matrix1[i][j] = scanner.nextInt();

}

}

System.out.println("Enter the elements of matrix2:"); for (int i = 0; i < rows2; i++) {

for (int j = 0; j < cols2; j++) { matrix2[i][j] = scanner.nextInt();

}

}

int[][] proOutput:duct = new int[rows1][cols2]; for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

for (int k = 0; k < cols1; k++) {

product[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

}

System.out.println("The product of the matrices is:"); for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) { System.out.print(product[i][j] + " ");

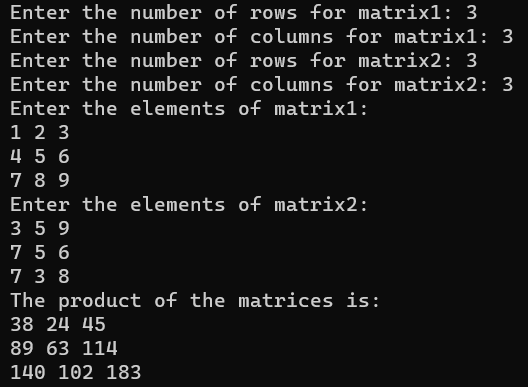
}

System.out.println();

}

}

}

**Output:**

# Vectors and Multithreading

* 1. Write a java program to implement the vectors.

**CODE:**

import java.util.Vector; class Vectors

{

public static void main(String args[])

{

Vector<String> v=new Vector<String>(); v.add("Yellow");

v.add("Purple");

v.add("Red");

System.out.println("Vector Elements are:-"+v); v.add(2,"Yellow");

System.out.println("After Adding Element at second position:-"+v); System.out.println("Element at third position:-"+v.get(3)); System.out.println("First Element:-"+v.ﬁrstElement()); System.out.println("Last Element:-"+v.lastElement()); System.out.println("Is this vector empty?"+v.isEmpty());

}

}

#### OUTPUT:

* 1. Write a java program to implement thread life cycle.

#### CODE:

public class Main2{

public static void main(String args[]) { Thread t=Thread.currentThread(); System.out.println("Current thread:"+t); t.setName("my Thread");

System.out.println("After changing name"+t); try{

for(int n=5;n>0;n--){ System.out.println(n); Thread.sleep(5000);

}

}

catch(InterruptedException e){ System.out.println("main thread interrupted");

}

}

}

#### OUTPUT:

* 1. Write a java program to implement multithreading.

#### CODE:

public class SimpleMultiThreadDemo { public static void main(String[] args) {

Thread thread1 = new Thread(() -> { for (int i = 0; i < 5; i++) {

System.out.println("Thread 1 is running..."); try {

Thread.sleep(500);

} catch (InterruptedException e) {

/ do nothing

}

}

});

Thread thread2 = new Thread(() -> { for (int i = 0; i < 5; i++) {

System.out.println("Thread 2 is running..."); try {

Thread.sleep(500);

} catch (InterruptedException e) {

/ do nothing

}

}

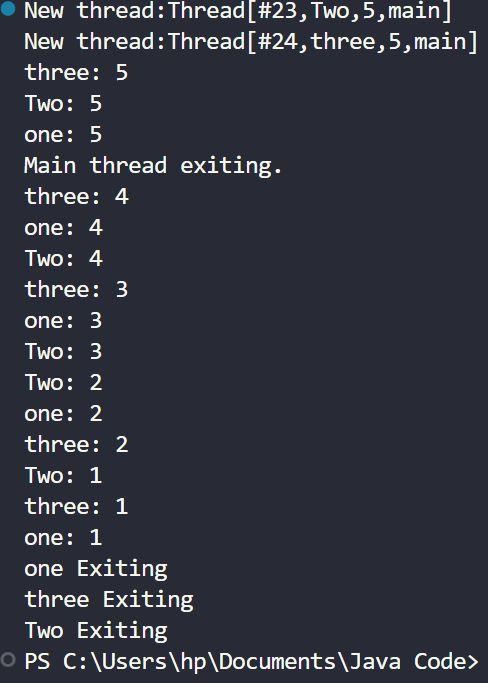
});

thread1.start(); thread2.start();

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

}

}

**OUTPUT:**

# File Handling

* 1. Write a java program to open a file and display the contents in the console window.

**Code:**

import java.io.BufferedReader; import java.io.FileReader; import java.io.IOException;

public class Main {

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

// Replace "filename.txt" with the name of your file FileReader fileReader = new FileReader("filename.txt");

BufferedReader bufferedReader = new BufferedReader(fileReader);

String line;

while ((line = bufferedReader.readLine()) != null) { System.out.println(line);

}

bufferedReader.close(); fileReader.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

#### Output:



* 1. Write a java program to copy the contents from one file to other file.

#### CODE:

import java.io.\*; public class Main {

public static void main(String[] args) { String sourceFile = "sourceFile.txt";

String destinationFile = "destinationFile.txt";

try (FileInputStream fileInputStream = new FileInputStream(sourceFile); FileOutputStream fileOutputStream = new FileOutputStream(destinationFile)) { byte[] buffer = new byte[1024];

int bytesRead;

while ((bytesRead = fileInputStream.read(buffer)) > 0) { fileOutputStream.write(buffer, 0, bytesRead);

}

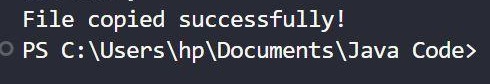
System.out.println("File copied successfully!");

} catch (IOException e) { e.printStackTrace();

}

}

}**Output:**



# Exception Handling

* 1. Write a java program to implement exception handling.

**Code:**

import java.util.Scanner; public class main5 {

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

{

int result = 10 / 0; System.out.println("Result: " + result);

}

catch (ArithmeticException e)

{

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

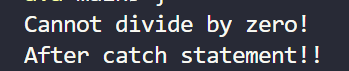
}

System.out.println("After catch statement!!");

}

}

#### Output



* 1. Write a java program to implement exception handling with multiple catch**. Code:**

import java.util.Scanner;

public class MultipleCatchExample { public static void main(String[] args) {

Scanner scanner = new Scanner(System.in); System.out.print("Enter a number: ");

try {

int num = scanner.nextInt(); int result = 10 / num;

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

}

catch (ArithmeticException e)

{

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

}

catch (NumberFormatException e)

{

System.out.println("Invalid number format!");

}

catch (Exception e)

{

System.out.println("Something went wrong!");

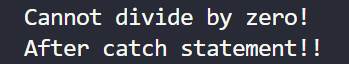
}

scanner.close();

}

}

**Output:**



# GUI Programming

## Construct a simple calculator using the JAVA awt with 3 text fields (the 3rd text field should be read-only) and 4 buttons to add, subtract, multiply, divide with minimum functionality.

**Code:**

import java.awt.\*; import java.awt.event.\*;

public class SimpleCalculator extends Frame implements ActionListener { TextField textField1, textField2, textField3;

Button addButton, subtractButton, multiplyButton, divideButton;

public SimpleCalculator() { setLayout(new GridLayout(4, 2)); setTitle("Simple Calculator");

textField1 = new TextField(); add(textField1);

Label labelPlus = new Label("+");

add(labelPlus);

textField2 = new TextField(); add(textField2);

Label labelEqual = new Label("="); add(labelEqual);

textField3 = new TextField(); textField3.setEditable(false); add(textField3);

addButton = new Button("Add"); addButton.addActionListener(this); add(addButton);

subtractButton = new Button("Subtract"); subtractButton.addActionListener(this); add(subtractButton);

multiplyButton = new Button("Multiply"); multiplyButton.addActionListener(this); add(multiplyButton);

divideButton = new Button("Divide"); divideButton.addActionListener(this); add(divideButton);

setSize(300, 150); setVisible(true);

}

public void actionPerformed(ActionEvent event) { if (event.getSource() == addButton) {

double num1 = Double.parseDouble(textField1.getText()); double num2 = Double.parseDouble(textField2.getText()); double result = num1 + num2; textField3.setText(Double.toString(result));

} else if (event.getSource() == subtractButton) {

double num1 = Double.parseDouble(textField1.getText()); double num2 = Double.parseDouble(textField2.getText()); double result = num1 - num2; textField3.setText(Double.toString(result));

} else if (event.getSource() == multiplyButton) {

double num1 = Double.parseDouble(textField1.getText()); double num2 = Double.parseDouble(textField2.getText());

double result = num1 \* num2; textField3.setText(Double.toString(result));

} else if (event.getSource() == divideButton) {

double num1 = Double.parseDouble(textField1.getText()); double num2 = Double.parseDouble(textField2.getText()); double result = num1 / num2; textField3.setText(Double.toString(result));

}

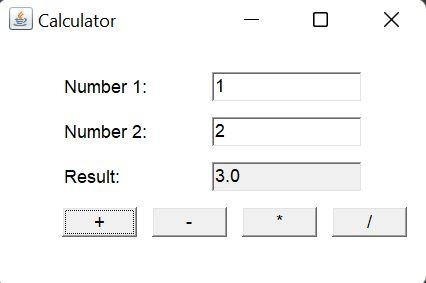
}

public static void main(String[] args) { SimpleCalculator calculator = new SimpleCalculator();

}

}

## Output:



###### Create an application using Java awt that accepts Principal Amount, No. of Years & Rate of Interest from 3 text ﬁelds, when you click the “Calculate

**Interest” button, the calculated interest should be displayed in a readonly text ﬁeld. When you click on the “Final Amount” button, the ﬁnal amount by adding principal amount and interest should also be displayed in a read-only text ﬁeld.**

**Code:**

import java.awt.\*; import java.awt.event.\*;

public class InterestCalculator extends Frame implements ActionListener { TextField principalField, yearsField, rateField, interestField, ﬁnalAmountField; Button calculateButton, ﬁnalAmountButton;

public InterestCalculator() { setLayout(new GridLayout(6, 2)); setTitle("Interest Calculator");

Label principalLabel = new Label("Principal Amount:"); add(principalLabel);

principalField = new TextField(); add(principalField);

Label yearsLabel = new Label("No. of Years:"); add(yearsLabel);

yearsField = new TextField(); add(yearsField);

Label rateLabel = new Label("Rate of Interest:"); add(rateLabel);

rateField = new TextField(); add(rateField);

Label interestLabel = new Label("Calculated Interest:"); add(interestLabel);

interestField = new TextField(); interestField.setEditable(false); add(interestField);

calculateButton = new Button("Calculate Interest"); calculateButton.addActionListener(this); add(calculateButton);

Label ﬁnalAmountLabel = new Label("Final Amount:"); add(ﬁnalAmountLabel);

ﬁnalAmountField = new TextField(); ﬁnalAmountField.setEditable(false); add(ﬁnalAmountField);

ﬁnalAmountButton = new Button("Final Amount"); ﬁnalAmountButton.addActionListener(this); add(ﬁnalAmountButton);

setSize(400, 200); setVisible(true);

}

public void actionPerformed(ActionEvent event) { if (event.getSource() == calculateButton) {

double principal = Double.parseDouble(principalField.getText()); double years = Double.parseDouble(yearsField.getText()); double rate = Double.parseDouble(rateField.getText());

double interest = (principal \* years \* rate) / 100.0; interestField.setText(Double.toString(interest));

} else if (event.getSource() == ﬁnalAmountButton) {

double principal = Double.parseDouble(principalField.getText()); double interest = Double.parseDouble(interestField.getText());

double ﬁnalAmount = principal + interest;

ﬁnalAmountField.setText(Double.toString(ﬁnalAmount));

}

}

public static void main(String[] args) {

InterestCalculator calculator = new InterestCalculator();

}

}

#### Output:

* 1. **Design an AWT program to print the factorial for an input value.**

#### CODE:

import java.awt.\*; import java.awt.event.\*;

import java.math.BigInteger;

public class FactorialCalculator extends Frame implements ActionListener { Label inputLabel, resultLabel;

TextField inputField, resultField; Button calculateButton;

public FactorialCalculator() { setTitle("Factorial Calculator"); setSize(300, 150);

setLayout(new GridLayout(3, 2)); inputLabel = new Label("Enter a number:"); inputField = new TextField();

resultLabel = new Label("Factorial:"); resultField = new TextField(); resultField.setEditable(false); calculateButton = new Button("Calculate"); add(inputLabel);

add(inputField); add(resultLabel); add(resultField); add(new Label("")); add(calculateButton);

calculateButton.addActionListener(this); setVisible(true);

}

public void actionPerformed(ActionEvent e) { if (e.getSource() == calculateButton) {

try {

int num = Integer.parseInt(inputField.getText()); BigInteger result = BigInteger.valueOf(1);

for (int i = 1; i <= num; i++) {

result = result.multiply(BigInteger.valueOf(i));

}

resultField.setText(result.toString());

} catch (NumberFormatException ex) { resultField.setText("Invalid input!");

}

}

}

public static void main(String[] args) { new FactorialCalculator();

}

}

#### OUTPUT: