**TITLE:** CodTech IT Solutions Internship

**Task Documentation:** “SIMPLE CALCULATOR” Using Java Programing.

**INTERN INFORMATION:**

**Name:** Nikhil Thipparthi

**ID:** ICOD6447

**INTRODUCTION**

A simple calculator is a fundamental tool that allows users to perform basic arithmetic operations quickly and efficiently. Whether it's adding up grocery expenses or calculating monthly budgets, a simple calculator serves as an essential aid in everyday calculations. In this introduction, we'll explore the significance, functionality, and importance of a simple calculator in various contexts.

**Implementation**

To implement a simple calculator in Java, we'll focus on the basic arithmetic operations: addition, subtraction, multiplication, and division.

* The program displays a menu with options for addition, subtraction, multiplication, division, and exiting.
* It prompts the user to choose an operation and then inputs the required numbers for the chosen operation.
* The corresponding method is called to perform the operation, and the result is displayed.

**CODE EXPLAINATION**

**JAVA CODE:**

import java.util.Scanner;

public class Calculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int choice;

do {

// Display menu

System.out.println("Choose an operation:");

System.out.println("1. Addition");

System.out.println("2. Subtraction");

System.out.println("3. Multiplication");

System.out.println("4. Division");

System.out.println("5. Exponentiation");

System.out.println("6. Quit");

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

// Get user choice

choice = scanner.nextInt();

// Perform operation based on user choice

switch (choice) {

case 1:

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

double addNum1 = scanner.nextDouble();

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

double addNum2 = scanner.nextDouble();

System.out.println("Result: " + add(addNum1, addNum2));

break;

case 2:

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

double subNum1 = scanner.nextDouble();

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

double subNum2 = scanner.nextDouble();

System.out.println("Result: " + subtract(subNum1, subNum2));

break;

case 3:

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

double mulNum1 = scanner.nextDouble();

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

double mulNum2 = scanner.nextDouble();

System.out.println("Result: " + multiply(mulNum1, mulNum2));

break;

case 4:

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

double divNum1 = scanner.nextDouble();

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

double divNum2 = scanner.nextDouble();

if (divNum2 != 0) {

System.out.println("Result: " + divide(divNum1, divNum2));

} else {

System.out.println("Error: Division by zero!");

}

break;

case 5:

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

double base = scanner.nextDouble();

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

double exponent = scanner.nextDouble();

System.out.println("Result: " + power(base, exponent));

break;

case 6:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice! Please enter a number between 1 and 6.");

}

} while (choice != 6);

scanner.close();

}

public static double add(double num1, double num2) {

return num1 + num2;

}

public static double subtract(double num1, double num2) {

return num1 - num2;

}

public static double multiply(double num1, double num2) {

return num1 \* num2;

}

public static double divide(double num1, double num2) {

return num1 / num2;

}

public static double power(double base, double exponent) {

return Math.pow(base, exponent);

}

}

**CODE EXPLANATION:**

1. **Import Statements:**

* import java.util.Scanner;: Imports the Scanner class from the Java utility package, which is used to take input from the user.

1. **Class Declaration:**

* public class SimpleCalculator {: Declares a public class named SimpleCalculator.

1. **Main Method:**

* public static void main(String[] args) {: The main entry point of the program.
* Scanner scanner = new Scanner(System.in);: Creates a new Scanner object to read user input from the console.

1. **Menu Display and User Input:**

* The program displays a menu of operations and prompts the user to enter their choice.
* The user's choice is stored in the variable choice.

1. **Switch Statement:**

* switch (choice) {: Based on the user's choice, the program executes a corresponding case block.
* Each case represents one of the operations: addition, subtraction, multiplication, division, or exit.

1. **Operation Methods:**

* Separate methods are defined for each arithmetic operation (addition, subtraction, multiplication, division).
* These methods take a Scanner object as an argument to read user input for the respective operation.

1. **Performing Arithmetic Operations:**

* Inside each operation method, the program prompts the user to input the required numbers and performs the respective operation.
* The result of the operation is returned to the main method.

1. **Displaying the Result:**

* After performing the operation, the result is displayed to the user.

1. **Closing Scanner:**

* scanner.close();: Closes the Scanner object to release system resources after the program finishes execution.

**USAGE**

We use calculators for various reasons, including:

* **Basic Arithmetic:** Calculators are handy tools for performing basic arithmetic operations such as addition, subtraction, multiplication, and division. They provide quick and accurate results, saving time and effort.
* **Complex Calculations**: Calculators can handle complex mathematical calculations that might be difficult or time-consuming to perform manually. This includes operations involving exponents, square roots, logarithms, trigonometric functions, and more.
* **Convenience:** Calculators offer convenience, especially in situations where precise calculations are required on the spot. They are commonly used in fields such as finance, engineering, science, and education.
* **Accuracy:** Calculators ensure accuracy in calculations, reducing the risk of human error compared to manual calculations. This is particularly important in critical applications where precision is essential.
* **Problem Solving:** Calculators aid in problem-solving by providing numerical solutions to mathematical problems. They are valuable tools for students, professionals, and individuals working in various industries.
* **Efficiency:** Calculators improve efficiency by simplifying complex calculations and automating repetitive tasks. This allows users to focus on higher-level analysis and decision-making.

**CONCLUSION**

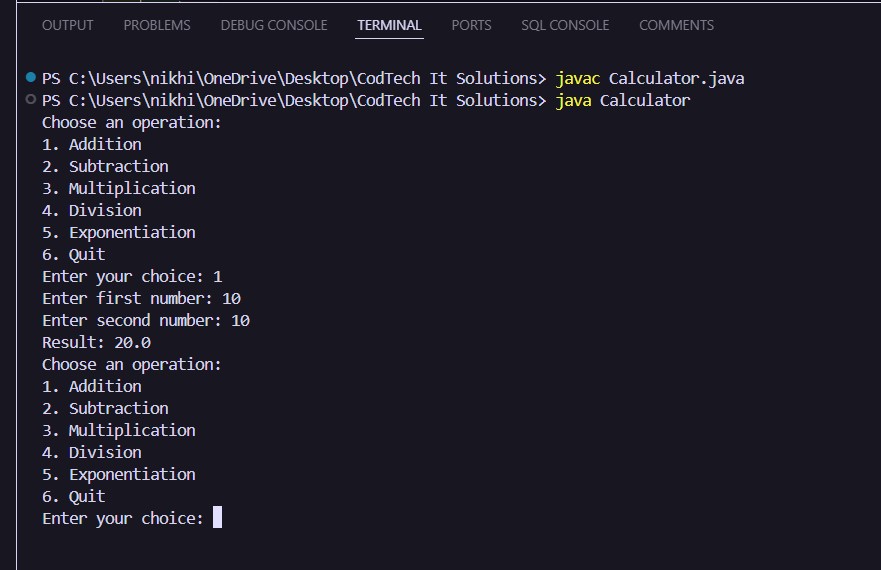
The simple calculator program built in Java provides a straightforward yet effective solution for performing basic arithmetic operations. Here's a brief overview of its key features and functionality:

* User-Friendly Interface: The calculator offers a menu-driven interface, allowing users to choose operations easily.
* Menu Options: Users can select from various arithmetic operations, including addition, subtraction, multiplication, division, and exponentiation.
* Input Handling: The program prompts users to input numbers for each operation, ensuring flexibility and customization.
* Error Handling: Error handling mechanisms are in place to handle invalid inputs or operations, such as division by zero.
* Efficiency: The program is designed to perform calculations quickly and accurately, providing instant results to the user.
* Reusable Code: Methods are utilized to encapsulate the logic for each arithmetic operation, promoting code reusability and maintainability.
* Looping Structure: The program utilizes a loop structure to allow users to perform multiple calculations within a single session, enhancing usability.

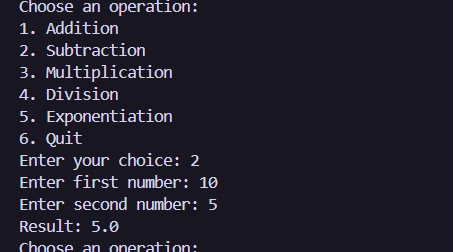
Overall, the simple calculator program offers a practical and intuitive tool for performing basic mathematical computations in a Java environment. It serves as a foundational example for understanding programming concepts such as user input, conditional statements, loops, and method definitions while providing a useful utility for everyday tasks.

**OUTPUT**

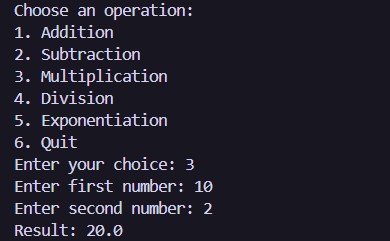
**Addition:**



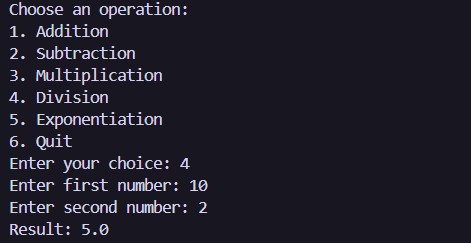
**Subtraction:**



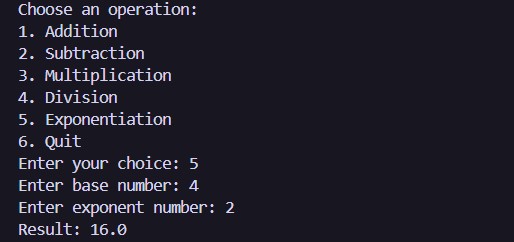
**Multiplication:**



**Division:**



**Exponentiation:**



**Quit:**

