

Perform an Operation Based on a Calculator Key

Problem Statement

In a calculator, when a user presses an operator key like '+' or '*', the calculator performs a specific arithmetic operation between two numbers.

In this task, you are given:

- Two numeric operands: a and b
- An operator key: op (which could be +, -, *, /)

A helper method named operate(double a, double b, char op) is already defined. It performs the correct operation based on the character provided.

Your job is to write just one line of code to call this method and print the result.

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
Test Case 1	10 4 /	2.5	2.5	Passed

The due date for submitting this assignment has passed.

1 out of 1 tests passed.

You scored 100.0/100.

Assignment submitted on 2025-04-16, 23:32 IST

Your last recorded submission was :

```
1 import java.util.Scanner;
2
3 public class W12_P3 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         // Read two numeric values
8         double a = sc.nextDouble();
9         double b = sc.nextDouble();
10
11        // Read the operator (e.g., '+', '-', '*', '/')
12        char op = sc.next().charAt(0);
13        System.out.print(operate(a, b, op));
14        sc.close();
15    }
16
17    // Helper method to perform basic arithmetic operations
18    public static double operate(double a, double b, char op) {
19        switch (op) {
20            case '+': return a + b;
21            case '-': return a - b;
22            case '*': return a * b;
23            case '/': return a / b;
24            default: return 0.0;
25        }
26    }
27 }
```

Sample solutions (Provided by instructor)

```
1 import java.util.Scanner;
2
3 public class W12_P3 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         // Read two numeric values
8         double a = sc.nextDouble();
9         double b = sc.nextDouble();
10
11        // Read the operator (e.g., '+', '-', '*', '/')
12        char op = sc.next().charAt(0);
13        System.out.println(operate(a, b, op)); // Call the operate method with inputs and print the result
14    }
15    /*
16    Explanation:
17    - The method operate(a, b, op) applies the given operation ('+', '-', '*', '/') on the two operands.
18    - It returns the result, which we directly print using System.out.println.
19    */
20    Final Note:
21    This task emphasizes how method calls can encapsulate logic and make code reusable and readable.
22    You have now practiced passing parameters, handling return values, and using methods effectively.
23    Completing Week 12 is a significant accomplishment, and we encourage you to reflect on how much progress you have made.
24    Stay curious and continue building upon this strong foundation in Java.
25    */
26    sc.close();
27    }
28
29    // Helper method to perform basic arithmetic operations
30    public static double operate(double a, double b, char op) {
31        switch (op) {
32            case '+': return a + b;
33            case '-': return a - b;
34            case '*': return a * b;
35            case '/': return a / b;
36            default: return 0.0;
37        }
38    }
39 }
```