Solution Review: Design a Calculator

In this lesson, you will see the detailed solution review of the challenge given in the previous lesson.

We'll cover the following

Solution

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

double add(double number1, double number2) {
  double result = number1 + number2;
  return result;
}

double subtract(double number1, double number2) {
  double result = number1 - number2;
  return result;
}

double multiply(double number1, double number2) {
  double result = number1 * number2;
  return result;
}

double divide(double number1, double number2) {
  double result = number1 / number2;
  return result;
}

double test(double number1, char operate, double number2) {
  double result;

  switch (operate) {

  case '+':
    result = add(number1, number2);
    break;

  case '-':
    result = subtract(number1, number2);
    break;

  case '*':
    result = multiply(number1, number2);
    break;

  case '/':
    result = divide(number1, number2);
    break;

  default:
    result = -1;
  }

  return result;

}

int main() {
  double number1 = 7;
  char operate = '+';
  double number2 = 8;
  double result;
  cout << number1 << operate << number2 << " = ";

  result = test(number1, operate, number2);
  cout << result;
  return 0;
}
```

Explanation

1- `add` function

The `add` function takes two values of type `double` in its input parameters, adds `number1` in `number2`, and returns the result of type `double` in the output.

double add (double number1 , double number2) ;

2- `subtract` function

The `subtract` function takes two values of type `double` in its input parameters, subtracts the `number2` from `number1` and returns the result of type `double` in the output.

```
double  subtract  ( double number1  ,  double number2 ) ;
```

3- `multiply` function

The `multiply` function takes two values of type `double` in its input parameters, multiplies the `number1` by `number2` and returns the result of type `double` in the output.

```
double  multiply   ( double number1  ,  double number2 ) ;
```

4- `divide` function

The `divide` function takes two values of type `double` in its input parameters, divides the `number1` by `number2`, and returns the result of type `double` in the output.

```
double  divide   ( double number1  ,  double number2 ) ;
```

`test` function

The function `test` takes two values of type `double` and one value of type `char` in its input parameters.

- `number1` and `number2` takes the values of the operands.
- `operate` can take $+$, $-$, $*$, and $/$ in its value.
    - If the value of `operate` is $+$, then it calls the `add` function and stores the output of the function in the `result`.
    - If the value of `operate` is $-$, then it calls the `subtract` function and stores the output of the function in the `result`.
    - If the value of `operate` is $*$, then it calls the `multiply` function and store the output of the function in the `result`.

  - If the value of `operate` is $/$, then it calls the `divide` function and store the output of the function in the `result`.
  - For any other value of `operate`, it should return **-1** in the output.

Let’s wrap up this chapter by solving a quiz in the upcoming lesson.