**C++ Program to Add Complex Numbers by Passing Structure to a Function**

This program takes two complex numbers as structures and adds them with the use of functions.

To understand this example, you should have the knowledge of the following [C++ programming](https://www.programiz.com/cpp-programming) topics:

* [C++ Structures](https://www.programiz.com/cpp-programming/structure)
* [C++ Structure and Function](https://www.programiz.com/cpp-programming/structure-function)

**Example: Source Code to Add Two Complex Numbers**

// Complex numbers are entered by the user

#include <iostream>

using namespace std;

typedef struct complex {

float real;

float imag;

} complexNumber;

complexNumber addComplexNumbers(complex, complex);

int main() {

complexNumber num1, num2, complexSum;

char signOfImag;

cout << "For 1st complex number," << endl;

cout << "Enter real and imaginary parts respectively:" << endl;

cin >> num1.real >> num1.imag;

cout << endl

<< "For 2nd complex number," << endl;

cout << "Enter real and imaginary parts respectively:" << endl;

cin >> num2.real >> num2.imag;

// Call add function and store result in complexSum

complexSum = addComplexNumbers(num1, num2);

// Use Ternary Operator to check the sign of the imaginary number

signOfImag = (complexSum.imag > 0) ? '+' : '-';

// Use Ternary Operator to adjust the sign of the imaginary number

complexSum.imag = (complexSum.imag > 0) ? complexSum.imag : -complexSum.imag;

cout << "Sum = " << complexSum.real << signOfImag << complexSum.imag << "i";

return 0;

}

complexNumber addComplexNumbers(complex num1, complex num2) {

complex temp;

temp.real = num1.real + num2.real;

temp.imag = num1.imag + num2.imag;

return (temp);

}

**Output**

Enter real and imaginary parts respectively:

3.4

5.5

For 2nd complex number,

Enter real and imaginary parts respectively:

-4.5

-9.5

Sum = -1.1-4i

In this program, two complex numbers entered by the user are stored in the structures num1 and num2.

These two structures are passed to addComplexNumbers() function which calculates the sum and returns the result to the main() function.

This result is stored in the structure complexSum.

Then, the sign of the imaginary part of the sum is determined and stored in the char variable signOfImag.

// Use Ternary Operator to check the sign of the imaginary number

signOfImag = (complexSum.imag > 0) ? '+' : '-';

If the imaginary part of complexSum is positive, then signOfImag is assigned the value '+'. Else, it is assigned the value '-'.

We then adjust the value of complexSum.imag.

/// Use Ternary Operator to adjust the sign of the imaginary number

complexSum.imag = (complexSum.imag > 0) ? complexSum.imag : -complexSum.imag;

This code changes complexSum.imag to positive if it is found to be of negative value.

This is because if it is negative, then printing it along with signOfImag will give us two negative signs in the output.

So, we change the value to positive to avoid sign repetition.

After this, we finally display the sum.