

# Assignment and Compound Assignment Operator

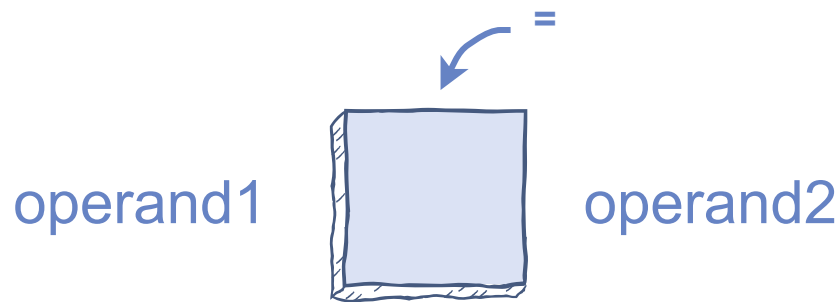
In this lesson, you will be introduced to the functionality of the assignment and the compound assignment operator.

## We'll cover the following ^

- Assignment operator
  - Example program
- Compound assignment operator
  - Example program

## Assignment operator #

The assignment operator takes the value on its right-hand side and assigns it to the operand on the left-hand side.



In C++, we have only one assignment operator.

Operator	Operation	Use
=	Assigns a value	Assigns a value of operand2 to operand1

## Example program #

The program given below shows the use of an assignment operator.

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

```
#include <iostream>
using namespace std;

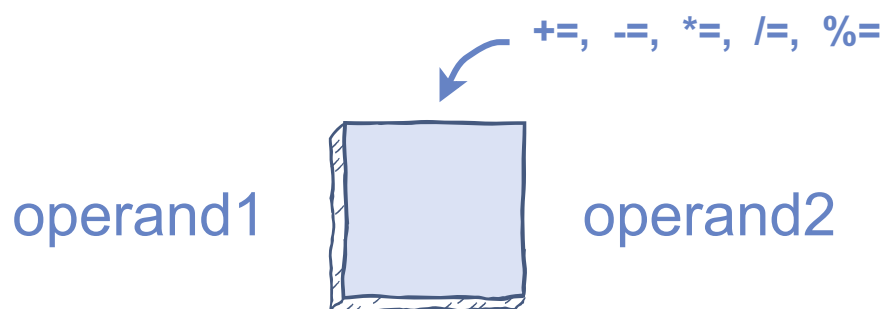
int main() {
    //Assigns value to the operands
    int operand1 = 50;
    float operand2 = 26;
    double operand3 = 78;
    bool operand4 = true;
    char operand5 = 'A';
    string operand6 = "Welcome";

    // Prints value of the operands
    cout << "operand1 = " << operand1 << endl;
    cout << "operand2 = " << operand2 << endl;
    cout << "operand3 = " << operand3 << endl;
    cout << "operand4 = " << operand4 << endl;
    cout << "operand5 = " << operand5 << endl;
    cout << "operand6 = " << operand6 << endl;
    return 0;
}
```



## Compound assignment operator #

The compound assignment operator is used to perform an operation, and then assign the result to the operand on the left-hand side.



Let's see the list of compound assignment operators available in C++.

Operator	Operation	Use
+=	Addition and assignment	Adds operand1 and operand2, and then assign updated value to operand1
-=	Subtraction and assignment	Subtracts operand2 from operand1, and then assign updated value to operand1
*=	Multiplication and assignment	Multiplies operand1 and operand2, and then assign updated value to operand1
/=	Division and assignment	Divides operand1 by operand2, and then assign updated value to operand1
%=	Modulus and assignment	Returns remainder after dividing operand1 by operand2, and then assign updated value to operand1

## Example program #

Consider two operands of type `int`: The value of `operand1` is `50`, and the value of `operand2` is `26`.

Run the code below to see the functionality of the compound assignment operator.

```
#include <iostream>
using namespace std;

int main() {
    // your code goes here
    int operand1 = 50;
    int operand2 = 26;
    cout << "Before using compound assignment operator:" << endl;
    cout << "operand1 = " << operand1 << endl;
    operand1 += operand2;
    cout << "After using compound assignment operator:" << endl;
    cout << "operand1 += operand2 = " << operand1 << endl;

    return 0;
}
```



Before using the assignment operator, the value of the `operand1` is `50`. The compound assignment operator `+=` first adds the value of `operand1` to the `operand2`. Then, it reassigns the new value to the `operand1`.



With this content, the following code is

What is the output of the following code?

```
int main() {  
    int operand1 = 50;  
    int operand2 = 26;  
    cout << "operand1 = " << operand1 << endl;  
    operand1 *= operand2;  
    cout << "operand1 *= operand2 = " << operand1 << endl;  
  
    return 0;  
}
```

[Retake Quiz](#)

---

Let's discuss the relational operators in the upcoming lesson.

Stay tuned!