

# OPERATÖRLER

Hazırlayan : Muhammed Esad Goncaloğlu

# OPERATÖRLER

- Operatörler, matematiksel, ilişkisel, bit düzeyinde, koşullu veya mantıksal manipölasyonları gerçekleştirmek için kullanılabilen sembollerdir.
- Operatörün derleyiciye belirli matematiksel, koşullu veya mantıksal işlevleri gerçekleştirmesini söyleyen bir sembol olduğunu da söyleyebiliriz.
- C de olduğu gibi C++ programlama dili de, programın ihtiyacına göre çeşitli görevleri yerine getirmek için birçok yerleşik operatöre sahiptir.

## Operators in C++

	Operator	Type
Unary operator	+ +, - -	Unary operator
Binary operator	+, -, *, /, %	Arithmetic operator
	<, <=, >, >=, ==, !=	Relational operator
	&&,   , !	Logical operator
	&,  , <<, >>, ~, ^	Bitwise operator
	=, +=, -=, *=, /=, %=	Assignment operator
Ternary operator	?:	Ternary or conditional operator

# ARİTMETİK OPERATÖRLER

- '+' operatörü, iki değişkeni toplar.
- '-' operatörü, iki değişkeni birbirinden çıkarır.
- '\*' operatörü, iki değişkeni birbiriyle çarpar.
- '/' operatörü, iki değişkeni birbirine böler.
- '%' operatörü, iki değişkenin böleninden kalanını hesaplar.
- '++' operatörü, kullanımına göre önce veya sonra bir artırır.
- '--' operatörü, kullanımına göre önce veya sonra bir azaltır.

```
#include <iostream>
using namespace std;
int main(){
    int num1=12,num2=3,result=0;
    result = num1 + num2; // 12 + 3 = 15
    cout << num1 << " + " << num2 << " = " << result << endl;
    result = num1 - num2; // 12 - 3 = 9
    cout << num1 << " - " << num2 << " = " << result << endl;
    result = num1 * num2; // 12 x 3 = 36
    cout << num1 << " x " << num2 << " = " << result << endl;
    result = num1 / num2; // 12 / 3 = 4
    cout << num1 << " / " << num2 << " = " << result << endl;
    result = num1 % num2; // 12 % 3 = 0
    cout << num1 << " % " << num2 << " = " << result << endl;
    return 0;
}
```

12 + 3 = 15

12 - 3 = 9

12 x 3 = 36

12 / 3 = 4

12 % 3 = 0

# BAĞLANTI OPERATÖRLER

- '=' operatörü, iki değişkenin birbirine eşit olup olmadığını kontrol eder.
- '!=' operatörü, iki değişkenin birbirine eşit değil olup olmadığını kontrol eder.
- '>' operatörü, iki değişkenin birbirinden büyük olup olmadığını kontrol eder.
- '<' operatörü, iki değişkenin birbirinden küçük olup olmadığını kontrol eder.
- '>=' operatörü, iki değişkenin birbirinden büyük veya eşit olup olmadığını kontrol eder.
- '<=' operatörü, iki değişkenin birbirinden küçük veya eşit olup olmadığını kontrol eder.

```
#include <iostream>
using namespace std;
int main(){
    int num1=16,num2=13;
    bool stat;
    stat = num1 == num2; // 16 == 13 false returns 0
    cout << num1 << " == " << num2 << " is " << stat << endl;
    stat = num1 != num2; // 16 != 13 true returns 1
    cout << num1 << " != " << num2 << " is " << stat << endl;
    stat = num1 > num2; // 16 > 13 true returns 1
    cout << num1 << " > " << num2 << " is " << stat << endl;
    stat = num1 < num2; // 16 < 13 true returns 0
    cout << num1 << " < " << num2 << " is " << stat << endl;
    stat = num1 >= num2; // 16 >= 13 true returns 1
    cout << num1 << " >= " << num2 << " is " << stat << endl;
    stat = num1 <= num2; // 16 <= 13 true returns 0
    cout << num1 << " <= " << num2 << " is " << stat << endl;
    return 0;
}
```

```
16 == 13 is 0
16 != 13 is 1
16 > 13 is 1
16 < 13 is 0
16 >= 13 is 1
16 <= 13 is 0
```

# ATAMA OPERATÖRLER

- '=' operatörü, iki değişkenin birbirine eşit olup olmadığını kontrol eder.
- '+=' operatörü, ilk değişken ile ikinci değişkeni toplar ve bu değeri ilk değişkene atar.
- '-=' operatörü, ilk değişken ile ikinci değişkenden çıkarır ve bu değeri ilk değişkene atar.
- '\*=' operatörü, ilk değişken ile ikinci değişkeni çarpar ve bu değeri ilk değişkene atar.
- '/=' operatörü, ilk değişkeni ikinci değişkene böler ve bu değeri ilk değişkene atar.
- '%=' operatörü, ilk değişkeni ikinci değişkene böler ve bu bölümden kalanı ilk değişkene atar.

```
#include <iostream>
using namespace std;
int main(){
    int num1=48,num2=17;
    // 48 += 17 = 63 num1 = 63
    cout << num1 << " += " << num2 << " is " << (num1+=num2) << endl;
    // 63 -= 17 = 48 num1 = 48
    cout << num1 << " -= " << num2 << " is " << (num1-=num2) << endl;
    // 48 *= 17 = 817 num1 = 817
    cout << num1 << " *= " << num2 << " is " << (num1*=num2) << endl;
    // 817 /= 17 = 48 num1 = 48
    cout << num1 << " /= " << num2 << " is " << (num1/=num2) << endl;
    // 48 %= 17 = 14 num1 = 14
    cout << num1 << " %= " << num2 << " is " << (num1%=num2) << endl;
    return 0;
}
```

```
48 += 17 is 65
65 -= 17 is 48
48 *= 17 is 816
816 /= 17 is 48
48 %= 17 is 14
```

# MANTIKSAL OPERATÖRLER

- ‘&&’ (VE) operatörü, bir veya birden fazla durumu inceler ve hepsinin doğruluğunu kontrol eder. Eğer hepsi doğru ise true (1) değer döndürür. Bir veya birden fazla durum yanlış ise false (0) döndürür.
- ‘||’ (VEYA) operatörü, bir veya birden fazla durumu inceler ve durumların doğruluğunu kontrol eder. Eğer bir durum bile doğru ise true (1) değer döndürür. Eğer bütün durumlar yanlış ise false (0) döndürür.

```
#include <iostream>
using namespace std;
int main(){
    bool stat;
    stat = 13 != 12 && 7 < -3; // 1 && 0 = 0
    cout << "13 != 12 && 7 < -3" << " is " << stat << endl;
    stat = 13 != 12 || 7 < -3; // 1 || 0 = 1
    cout << "13 != 12 || 7 < -3" << " is " << stat << endl;
    stat = 13 > 12 && 7 != -3; // 1 && 1 = 1
    cout << "13 > 12 && 7 != -3" << " is " << stat << endl;
    stat = 13 > 12 || 7 != -3; // 1 || 1 = 1
    cout << "13 > 12 && 7 != -3" << " is " << stat << endl;
    return 0;
}
```

```
13 != 12 && 7 < -3 is 0
13 != 12 || 7 < -3 is 1
13 > 12 && 7 != -3 is 1
13 > 12 && 7 != -3 is 1
```

# KAYNAKÇA

- [https://www.tutorialspoint.com/cplusplus/cpp\\_operators.htm](https://www.tutorialspoint.com/cplusplus/cpp_operators.htm)
- <https://www.geeksforgeeks.org/operators-in-c/?ref=lbp>