Float X Double

The precision of a floating point value indicates how many digits the value can have after the decimal point. The precision of float is only six or seven decimal digits, while double variables have a precision of about 15 digits. Therefore it is safer to use double for most calculations

Scientific Numbers

A floating point number can also be a scientific number with an "e" to indicate the power of 10:

```
#include <iostream>
using namespace std;
int main() {
    float f1 = 85e3;
    double d1 = 22E4;
    cout << f1 << "\n";
    cout << d1 ;
    return 0;
}
```

```
85000
220000
```

Booleans

A boolean data type is declared with the bool keyword and can only take the values true or false. When the value is returned, true = 1 and false = 0

```
#include <iostream>
using namespace std;
int main() {
  bool isCodingFun = true;
  bool isFishTasty = false;
  cout << isCodingFun << "\n";
  cout << isFishTasty;
  return 0;
}
```

```
1
0
```

Characters

The char data type is used to store a single character. The character must be surrounded by single quotes like "A" or "C" :

```
#include <iostream>
using namespace std;
int main() {
  char myGrade = 'B';
  cout << myGrade;
  return 0;
}
```

```
B
```

```
#include <iostream>
using namespace std;
int main(){
  char a=65, b=66, c=67;
  cout << a;
  cout << b;
  cout << c;
  return 0;
}
```

```
ABC
```

C++ Operators

Operators are used to perform operations on variables and values. The value is called an operand, while the operation (to be performed between the two operands) is defined by an operator:
(İşlenen)

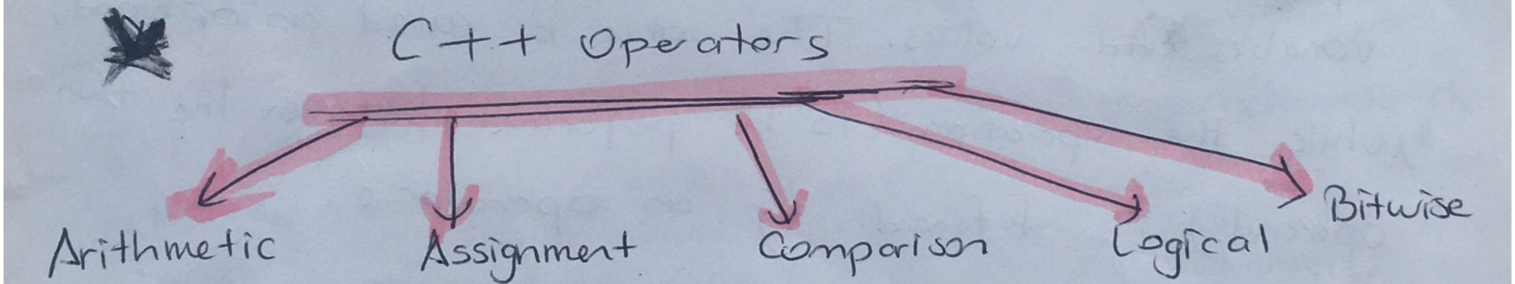
Operand	operator	operand
200	+	40

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

int main() {
  int x = 200 + 40;
  cout << x;
 return 0;
}
```

```
240
```

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

int main() {
  int sum1 = 100 +50;
  int sum2 = sum1 + 250;
  int sum3 = sum2 + sum2;
  cout << sum1 << "\n";
  cout << sum2 << "\n";
  cout << sum3;
  return 0;
}
```

```
150
400
800
```



Arithmetic Operators

Arithmetic Operators are used to perform common mathematical operations.

- **+ (Addition):**

Adds together two values. Ex: x+y

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

int main() {
    int x = 5;
    int y = 3;
  cout << x+y;
  return 0;
}
```

```
8
```

- (subtraction):

Subtracts one value from another   Ex : x-y

```
#include <iostream>
using namespace std;
int main() {
    int x = 50;
    int y = 30;
    cout << x-y;
    return 0;
}
```

20

* (multiplication):

Multiplies two values   x*y

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

int main() {
  int x = 5;
  int y = 3;
  cout << x*y;
  return 0;
}
```

15

/ (Division) :

Divides one value from another   x/y

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

int main() {
int x=12;
int y=4;
cout << x/y ;
return 0;
}
```

3

% Modulus => Bölme işleminden Kalanı verir.

Returns the division remainder

x % y

```
5 | 2
-4  2
 1
```

```
#include <iostream>
using namespace std;
int main() {
   int x=5 ;
   int y=2 ;
   cout << x % y;
   return 0;
}
```

1