**Operations in C++**

In C++, **operations** are the actions that can be performed on data values, known as **operands**, using specific **operators**. Operators define how operands behave during these operations, which can involve arithmetic, logic, assignment, comparison, etc.

**Types of Operators in C++**

C++ operators are broadly categorized into several types:

1. **Arithmetic Operators**
2. **Relational Operators**
3. **Logical Operators**
4. **Bitwise Operators**
5. **Assignment Operators**
6. **Increment/Decrement Operators**
7. **Conditional (Ternary) Operator**
8. **Special Operators**

**1. Arithmetic Operators**

These operators are used for basic mathematical operations.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| + | Addition | a + b |
| - | Subtraction | a - b |
| \* | Multiplication | a \* b |
| / | Division | a / b |
| % | Modulus (remainder) | a % b |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 10, b = 3;**

**cout << "Addition: " << a + b << endl;**

**cout << "Subtraction: " << a - b << endl;**

**cout << "Multiplication: " << a \* b << endl;**

**cout << "Division: " << a / b << endl;**

**cout << "Modulus: " << a % b << endl;**

**return 0;**

**}**

**2. Relational Operators**

Relational operators are used to compare two values. The result is either true or false.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| == | Equal to | a == b |
| != | Not equal to | a != b |
| > | Greater than | a > b |
| < | Less than | a < b |
| >= | Greater than or equal | a >= b |
| <= | Less than or equal | a <= b |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 5, b = 10;**

**cout << (a == b) << endl; // False, outputs 0**

**cout << (a != b) << endl; // True, outputs 1**

**cout << (a < b) << endl; // True, outputs 1**

**cout << (a > b) << endl; // False, outputs 0**

**return 0;**

**}**

**3. Logical Operators**

Logical operators are used to combine two or more conditions.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| && | Logical AND | (a && b) |
| ` |  | ` |
| ! | Logical NOT | (!a) |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**bool a = true, b = false;**

**cout << (a && b) << endl; // False, outputs 0**

**cout << (a || b) << endl; // True, outputs 1**

**cout << (!a) << endl; // False, outputs 0**

**return 0;**

**}**

**4. Bitwise Operators**

Bitwise operators are used to manipulate data at the bit level.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| & | Bitwise AND | a & b |
| ` | ` | Bitwise OR |
| ^ | Bitwise XOR | a ^ b |
| ~ | Bitwise NOT | ~a |
| << | Left shift | a << 2 |
| >> | Right shift | a >> 2 |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 5, b = 9;**

**cout << (a & b) << endl; // Bitwise AND**

**cout << (a | b) << endl; // Bitwise OR**

**cout << (a ^ b) << endl; // Bitwise XOR**

**cout << (a << 1) << endl; // Left shift**

**cout << (a >> 1) << endl; // Right shift**

**return 0;**

**}**

**5. Assignment Operators**

Assignment operators are used to assign values to variables.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| = | Assign value | a = b |
| += | Add and assign | a += b |
| -= | Subtract and assign | a -= b |
| \*= | Multiply and assign | a \*= b |
| /= | Divide and assign | a /= b |
| %= | Modulus and assign | a %= b |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 10;**

**a += 5; // a = a + 5**

**cout << a << endl; // Outputs 15**

**a \*= 2; // a = a \* 2**

**cout << a << endl; // Outputs 30**

**return 0;**

**}**

**6. Increment/Decrement Operators**

These operators are used to increase or decrease the value of a variable by 1.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| ++ | Increment by 1 | ++a |
| -- | Decrement by 1 | --a |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 5;**

**cout << ++a << endl; // Pre-increment: a becomes 6**

**cout << a++ << endl; // Post-increment: outputs 6, then a becomes 7**

**cout << a << endl; // Outputs 7**

**return 0;**

**}**

**7. Conditional (Ternary) Operator**

This is a shorthand for if-else statements.

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| ?: | Conditional (Ternary Operator) | a ? b : c |

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 10, b = 5;**

**int max = (a > b) ? a : b; // If a > b, assign a to max, else assign b**

**cout << "Max: " << max << endl;**

**return 0;**

**}**

**8. Special Operators**

* **sizeof**: Returns the size of a variable or datatype.
* **& (Address-of)**: Returns the memory address of a variable.
* \*\*\* (Pointer dereference)\*\*: Accesses the value at the address stored in a pointer.

**Example:**

cpp

**#include <iostream>**

**using namespace std;**

**int main() {**

**int a = 5;**

**cout << "Size of a: " << sizeof(a) << endl;**

**cout << "Address of a: " << &a << endl;**

**int\* ptr = &a;**

**cout << "Value at ptr: " << \*ptr << endl;**

**return 0;**

**}**

These operators are fundamental in C++ and allow you to manipulate variables and values in various ways to perform tasks like arithmetic, comparisons, logic operations, and memory handling.