

newVariables

Variables in C++ are used to store data values. Each variable must have a specific type (e.g., int, double, char, string, etc.), which determines what kind of data it can hold that is stored in memory.

Example:

```
int val = 5;  
double d = 2.0;  
string s = "ABC";  
char c = 'A';
```

Expressions

Expressions combine variables, constants, and operators to compute values.

Example:

```
int a = 10;  
int b = 5;  
int sum = a + b; // Expression: addition
```

Arithmetic and logical operators

Arithmetic Operators:

These perform mathematical operations on variables and values.

+ : Addition

- : Subtraction

* : Multiplication

/ : Division

% : Modulus (remainder after division)

Examples:

```
int result = (a + b) * 2; // Arithmetic expression of addition and multiplication
```

```
int result2 = 4%2; calculates the remainder when 4 is divided by 2 so it's 0
```

Precedence rules. When and when not to use parentheses.

Operator precedence determines the order in which operators are evaluated in expressions.

Example:

```
int x = 5 + 2 * 3; // Multiplication is done first, so its 3*2 which is equal to 6 and then adds 5, x = 11
```

```
int y = (5 + 2) * 3; // Parentheses force addition first, so its 5+2 which is 7 then multiplied by 3, y = 21
```

Simple input and output (using std::cin and std::cout)

For console input and output, C++ uses std::cin for input and std::cout for output.

Output with std::cout:

Use std::cout to print messages or values to the console.

You can also use this line of code at the top of your program **using namespace std;** this makes it so that you don't have to use `std::` for the beginning of each line of code

Examples:

Using namespace std

```
Int main(){
    cout << "Hello, World!" << std::endl; // Outputs: Hello, World!
}
```

Using namespace std

```
Int main(){
    int age;
    cout << "Enter your age: ";
    cin >> age; // Stores the input in the 'age' variable
}
```

File input/output:

C++ allows you to read from and write to files using the <fstream> library.

Steps:

Include the header <fstream>.

Create an input stream (std::ifstream) to read from a file.

Create an output stream (std::ofstream) to write to a file.

Example:

```
#include <iostream>
```

```
#include <fstream> // For file operations
```

```
Using namespace std;
```

```
int main() {
```

```
    // Writing to a file
```

```
    ofstream outfile("output.txt");
```

```
    if (outfile.is_open()) {
```

```
        outfile << "Hello, file!" << endl;
```

```
        outfile.close(); // Always close the file when done
```

```
    }
```

```
    // Reading from a file
```

```
    ifstream infile("output.txt");
```

```
    string line;
```

```
    if (infile.is_open()) {
```

```
        while (getline(infile, line)) {
```

```
            cout << line << endl; // Output file content to console this prints every line in the
```

```
file
```

```
        }
```

```
        infile.close(); // Close the file
    }

    return 0;
}
```