

C++ Basics

- Viraj Chandra

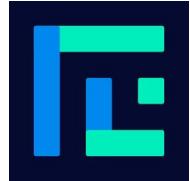


Goal

To understand:

- Constants, variables and datatypes
- Input / Output
- Different types of operators
- Conditional statements

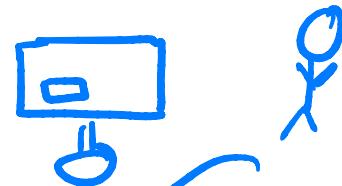
We will be able to write simple programs by the end using conditional statements and arithmetic operators, all in C++ language. (eg. Assigning grades based on marks)



Why you should prefer C++? (For CP)

- Efficiency and Speed
 - Generally, C++ > JAVA > Python in terms of execution time.
- Most popular language for CP
- In-built Data Structures and Algorithms (STL)

Simplest C++ program



```
1 ✓ #include <iostream>
2 ✗ using namespace std;
3
4 ✓ int main()
5 {
6     cout << "Hello world!" << endl;
7 }
```

input
↑↑ output

The code is annotated with handwritten text and arrows. The word "input" is written above the monitor, and "output" is written above the "cout" line in the code, with two arrows pointing from the text to the monitor and the "cout" line respectively. There are also three colored circles (red, yellow, green) at the top left of the code area.



Input in C++

To input a value, we use the `cin` operator as follows: `cin >> value;`

`2 3 4`

`int a`

To input multiple values in the same line:

`cin >> value1 >> value2 >> value3;`

`a
2 b
3 c
4`

NOTE: Each input value must be separated by a space or a new line.

`2 3 4`



Output in C++

To output a value, we use the cout operator as follows: `cout << value;`

To output multiple values in the same line:

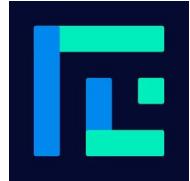
`cout << value1 << value2 << value3;`

a b c

abc

NOTE: To start printing in a new line: `endl` or '`\n`'

a b a
 b



Datatypes

Datatypes are used to set the “type” of a variable.

For example, int is used to declare integer variables.

Two types of datatypes:

- Primitive datatypes → int, short, long, double, float, char ...
- Derived datatypes → String, object ...



Common Primitive Datatypes

```
1 // Integer types
2 int a = 10;           // Regular integer
3 long long int b = 1000000; // Larger integer / long long / long
4 unsigned int c = 42;    // Non-negative integer
5
6 // Character type
7 char d = 'A';         // Single character
8
9 // Boolean type
10 bool e = true;        // Boolean (true/false)
11
12 // Floating-point types
13 float f = 3.14f;      // Single-precision floating-point
14 double g = 3.1415926535; // Double-precision floating-point
15 long double h = 3.141592653589793238L; // Extended-precision floating-point
16
17 // Special type
18 void* ptr = nullptr;   // Void pointer (points to no type)
19
```



int v/s long long int

$1e9 = 10^9$

$-2 \times 10^9 \quad 2 \times 10^9$

int can store integers from $-2e9$ to $2e9$

$-9 \times 10^{18} \quad 9 \times 10^{18}$

long long int can store integers from $-9e18$ to $9e18$

Using long long int doesn't hurt, but using int might give you wrong answer if values are too big.

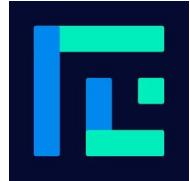
You can also use **long long instead of long long int.**



Common Derived Datatypes

- ✓ string
- ✓ vector
- ✓ map
- ✓ set
- ✓ priority_queue
- ✓ object

In depth explanation of these datatypes with syntax will be covered in
the upcoming classes.



Constants in C++

- Integer constants: 4 , 62 , -90
- Decimal constants: 3.14 , 12.0 , 0.33333
- Character constants: 'f' , '5' , '~' , '\n'
- String literal: "Hello :D" , "MyP@ssw0rd123!"

Const int a = 5



Variables

Variables are containers that stores specific types of data. They can be modified with the assignment operator “=”

Syntax:- **datatype variable_name = value;**

int a = 10;



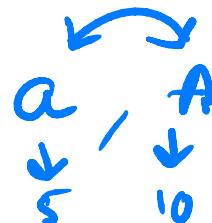
Variables

viraj_chandna

Variable names cannot:

- Have spaces (use underscore instead)
- Start with a digit ✗ 9a , a9 ✓
- Be reserved by the compiler (Keywords not allowed) int
- Already taken by another variable (in the same scope)

NOTE: Variables are case sensitive





Unary Operators in C++

 $\rightarrow \underline{a = 5}$

Operators that only need one value/operand are called unary operators.

- $+$
- $-$
- $++$
- $--$

$a++ \quad a=5 \rightarrow 6$
 $a-- \quad a=6 \rightarrow 5$

Pre-before

, $++a$ ↗ increment a , then use it

Post-after

, $a++$ ↗ use a , then increment it



Arithmetic Operators in C++

- ✓ • + Addition
- ✓ • - Subtraction
- ✓ • * Multiplication
- ✓ • / Division (Quotient)
- ✓ • % Modulo (Remainder)

$$a \% b = 0 \\ 6 \quad 3$$

NOTE: C++ follows the BODMAS rule, for example ->

int result = (2 + 3) * 4 - 5; // Brackets first: (2+3)=5, then
multiplication: 5*4=20, then subtraction: 20-5=15



Check Your Understanding

①

char ch = '!'

How will you declare a **character** equal to exclamatory mark?

②

Take two values a, b as **input**, and **output** three values:

a+b and a*b and a/b

`cin >> a >> b;`

`cout << a+b << ' ' << a*b << ' ' << a/b;`

③

a/b should be a decimal, not an integer

`a = 5`

`b = 3`

`c = a/b`

`cout << c << endl;`

a

b

int

int

a/b

c

double





Conditions and Relational Operators in C++

Conditions return a boolean value depending on whether the expression is true or false.

Conditional operators:

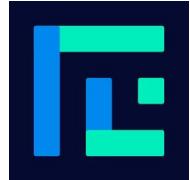
`==, !=` → **not equal**

↓
is equal

Relational operators:

`<, >, <=, >=`

↑ ↑ ↑ ↑



Logical Operators in C++

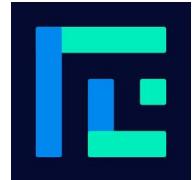
Logical operators perform operations on boolean values or expressions that result in Boolean values.

(**a == b** **c == d**)
“(expr1) && (expr2)” checks whether **BOTH** are true.

(“(expr1) || (expr2)”) checks whether **EITHER one is true**.
“!(expr)” returns the **OPPOSITE** of the result of “expr”

The operators are called **AND, OR, NOT** operators respectively.



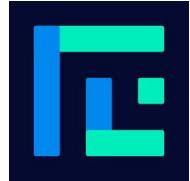


Conditional Statements

Conditional statements execute a different block of code depending on the boolean value of a condition.

Syntax:-

```
Expt = [a==b]
if (condition) {
    // something a=a+5
} else if (another_condition) {
    // something a<b
} else {
    // something
}
```



Challenge



Given someone's age, tell whether they are a child, adult, or a senior citizen.

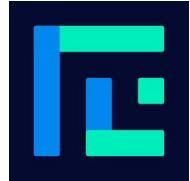
0 - 17 : Child

18 - 64 : Adult

65 + : Senior Citizen



Take input of 3 numbers x, y, z and output the maximum using if statements.



Resources

<https://www.programiz.com/cpp-programming> (learning C++ in general)

<https://www.programiz.com/cpp-programming/operators> (all operators)

https://www.w3schools.com/cpp/cpp_conditions.asp (operators, if-statements)