



# C++ BASICS (PART 1)

--by HARSH GUPTA

# Why you should prefer C++

(For Competitive Programming)

- Efficiency and Speed
- Most popular language for CP
- In-built Data Structures and Algorithms (STL)

too convincing

logic

code

90%

java

python

C++14

C++17

C++20

java

C++

2-3

10 lines

java

C++

offer

C++

java

C++

C++

# Goal

To understand:

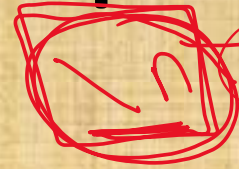
- Constants and datatypes
- Input/Output
- Different types of operators
- Conditional statements

C++

We will be able to write simple programs by the end using conditional statements and arithmetic operators (eg. A-F grade assigner)

# Simplest C++ program

the  
the



new line

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

int main() {
    cout << "Hello world!" << endl;
}
```

# Constants in C++

4      2      0

✓ • Integer constants: 4 | 62 | -90

✓ • Decimal constants: 3.14 | 12.0 | 0.33333

✓ • Character constants: 'f' | '5' | '~' | '\n'

✓ • String literal: "Hello :D" | "MyP@ssw0rd123!"





# Output in C++

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

To print multiple values in the same line:  
cout << value1 << value2 << value3;

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

# Arithmetic operators in C++

Arithmetic Operators:

- ✓ 1) + Addition  $\rightarrow 5 \times 2$
- ✓ 2) - Subtraction  $\rightarrow 5 - 2$
- ✓ 3) \* Multiplication  $\rightarrow 5 \times 2$
- ✓ 4) / Division (Quotient)  $\rightarrow 5 / 2$
- ✓ 5) % Modulo (Remainder)

$$5 \div 2 = 1$$

NOTE: C++ follows the BODMAS rule

floor division

$$5.0 \div 2.0$$

$$2.5$$

$$2.5 \rightarrow 2 \text{ (floor)}$$

$$6 \div 2 = 3$$

$$7 \div 2 = 3$$

# Variables

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

Syntax: `datatype variable_name = value;`



# Variables

Variable names cannot:

- Have spaces (use underscore instead)
- Start with a digit
- Be reserved by the compiler (Keywords not allowed)
- Already taken by ~~another~~ variable (in the same scope)

NOTE: Keywords/Variables are case sensitive

# Datatypes

int float  
double

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 ~~✓~~
- Derived datatypes



# Common Primitive datatypes

1. int (long long int, unsigned int, etc.)
2. char
3. bool
4. float (double, long double)
5. Special type: void

64 bits

64 bits

32 bits

sign

+ve/-ve

unsigned  
long long  
int

online

+

0

1 time

long long int → 64

unsigned long long int → 64

unsigned int → 0

int → 0

$$\text{floor}(x) = \lfloor x \rfloor$$

closest  
floor

$$\text{ceil}(5.3) = 6$$

# Common Derived datatypes

1. string
2. vector
3. map
4. set
5. priority\_queue

$$\text{ceil}(5) = 5$$

$$\text{ceil}(5.3) = 6$$

$$\text{ceil}(5.8) = 6$$

$$\text{ceil}(-5.2) = -5$$



# Arithmetic Assignment Operators

1. +=

2. -=

3. \*=

4. /=

5. %=

int C = 5      (15)

C ← (C + 10)

C = C + 10 ; ?

C += 10 ;

C -= 10 ; → C = C - 10

C \*= 10 ;

C = C \* 10



# Unary Operators

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

- 1. +

- 2. -

- 3. ++

- 4. --

~~C~~ C ++;  
C --;

C = C + 10  
↓  
binary  
operator

# Input in C++

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

To print multiple values in the same line:  
`cin >> value1 >> value2 >> value3;`

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

# Check your understanding - 1

skipping

H/W

1. 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  $a/b$

$a/b$  should be a decimal, not an integer

# Conditions and Relational Operators

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

Conditional operators:

`==, !=`

Relational operators:

`<, >, <=, >=`

# Logical operators

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

1. “(expr1) && (expr2)” checks whether BOTH are true.
2. “(expr1) || (expr2)” checks whether EITHER one is true.
3. “!(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:

```
if (condition) {  
    // something  
} else if (another_condition) {  
    // something  
} else {  
    // something  
}
```


# Check Your Understanding 2

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

0-17 : Child

18-64 : Adult

65+ : Senior Citizen

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

THANK YOU 😊