C++ Basics (Part 1)

Srivaths P

Why you should prefer C++
(For Competitive Programming)

- Efficiency and Speed

- Most popular language for CP

- In-built Data Structures and Algorithms (STL)

Goal

To understand:
- Constants 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 (eg. A-F grade assigner)

Simplest C++ program

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

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

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!"

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

2) - Subtraction

3) * Multiplication

4) / Division (Quotient)

5) % Modulo (Remainder)

NOTE: C++ follows the BODMAS rule

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

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

Common Derived datatypes

1. string
2. vector
3. map
4. set
5. priority_queue

Arithmetic Assignment Operators

1. +=
2. -=
3. *=
4. /=
5. %=

Unary Operators

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

1. +
2. -
3. ++
4. --

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

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 (operators, if-statements)

Thank you!