### **Variables in C++:**

In programming, a variable is a container (storage area) to hold data.

To indicate the storage area, each variable should be given a unique name (identifier). For example,

int age = 14;

Here, age is a variable of the int data type, and we have assigned an integer value 14 to it.

**Note:** The int data type suggests that the variable can only hold integers. Similarly, we can use the Double data type if we have to store decimals and exponentials.

### **Literals in C++:**

Literals are data used for representing fixed values. They can be used directly in the code.

For example: 1, 2.5, 'c' etc.

Here, 1, 2.5 and 'c' are literals. Why? You cannot assign different values to these terms.

Here's a list of different literals in C++ programming.

1. Integers:

An integer is a numeric literal(associated with numbers) without any fractional or exponential part. There are three types of integer literals in C programming:

* decimal (base 10)
* octal (base 8)
* hexadecimal (base 16)

For example:

Decimal: 0, -9, 22 etc

Octal: 021, 077, 033 etc

Hexadecimal: 0x7f, 0x2a, 0x521 etc

In C++ programming, octal starts with a 0, and hexadecimal starts with a 0x.

1. Floating Point Literals:

A floating-point literal is a numeric literal that has either a fractional form or an exponent form. For example:

-2.0

0.0000234

-0.22E-5

**Note:** E-5 = 10-5

1. Characters:

A character literal is created by enclosing a single character inside single quotation marks. For example: 'a', 'm', 'F', '2', '}' etc.

1. Escape Sequences:

Sometimes, it is necessary to use characters that cannot be typed or has special meaning in C++ programming. For example, newline (enter), tab, question mark, etc.

1. String Literals:

A string literal is a sequence of characters enclosed in double-quote marks. For example:

|  |  |
| --- | --- |
| "good" | string constant |
| "" | null string constant |
| " " | string constant of six white space |
| "x" | string constant having a single character |
| "Earth is round\n" | prints string with a newline |

### **Constants:**

We can create variables whose value cannot be changed. For that, we use the const keyword. Here's an example:

const int LIGHT\_SPEED = 299792458;

LIGHT\_SPEED = 2500 // Error! LIGHT\_SPEED is a constant.

Here, we have used the keyword const to declare a constant named LIGHT\_SPEED. If we try to change the value of LIGHT\_SPEED, we will get an error.

A constant can also be created using the #define preprocessor directive.

### **Data Types in C++**

In C++, data types are declarations for variables. This determines the type and size of data associated with variables.

For example, int age = 13;

Here, age is a variable of type int. Meaning, the variable can only store integers of either 2 or 4 bytes.

## **Fundamental Data Types:**

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Meaning** | **Size (in Bytes)** |
| Int | Integer | 2 or 4 |
| Float | Floating-point | 4 |
| Double | Double Floating-point | 8 |
| Char | Character | 1 |
| wchar\_t | Wide Character | 2 |
| Bool | Boolean | 1 |
| Void | Empty | 0 |

1. **int:**

* The int keyword is used to indicate integers.
* Its size is usually 4 bytes. Meaning, it can store values from **-2147483648 to 2147483647**.
* For example, int salary = 85000;

1. **float and double:**

* float and double are used to store floating-point numbers (decimals and exponentials).
* The size of float is 4 bytes and the size of double is 8 bytes. Hence, double has two times the precision of float. To learn more, visit C++ float and double.
* For example,

float area = 64.74;

double volume = 134.64534;

As mentioned above, these two data types are also used for exponentials.

For example,

double distance = 45E12 // 45E12 is equal to 45\*10^12

1. **char:**

* Keyword char is used for characters.
* Its size is 1 byte.
* Characters in C++ are enclosed inside single quotes ' '.
* For example,

char test = 'h';

**Note:** In C++, an integer value is stored in a char variable rather than the character itself.

1. **wchar\_t:**

* Wide character wchar\_t is similar to the char data type, except its size is 2 bytes instead of 1.
* It is used to represent characters that require more memory to represent them than a single char.
* For example,

wchar\_t test = L'ם' // storing Hebrew character;

Notice the letter L before the quotation marks.

**Note:** There are also two other fixed-size character types char16\_t and char32\_t introduced in C++11.

1. **bool:**

* The bool data type has one of two possible values: true or false.
* Booleans are used in conditional statements and loops (which we will learn in later chapters).
* For example,

bool cond = false;

1. **void:**

* The void keyword indicates an absence of data. It means "nothing" or "no value".
* We will use void when we learn about functions and pointers.

**Note:** We cannot declare variables of the void type.

### **Derived Data Types:**

Data types that are derived from fundamental data types are derived types.

For example: arrays, pointers, function types, structures, etc.

### **Type Modifiers:**

We can further modify some of the fundamental data types by using type modifiers. There are 4 type modifiers in C++. They are:

1. signed
2. unsigned
3. short
4. long

We can modify the following data types with the above modifiers:

* int
* double
* char

### **Modified Data Types List:**

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Size (in Bytes)** | **Meaning** |
| signed int | 4 | used for integers (equivalent to int) |
| unsigned int | 4 | can only store positive integers |
| Short | 2 | used for small integers (range **-32768 to 32767**) |
| Long | at least 4 | used for large integers (equivalent to long int) |
| unsigned long | 4 | used for large positive integers or 0 (equivalent to unsigned long int) |
| long long | 8 | used for very large integers (equivalent to long long int). |
| unsigned long long | 8 | used for very large positive integers or 0 (equivalent to unsigned long long int) |
| long double | 8 | used for large floating-point numbers |
| signed char | 1 | used for characters (guaranteed range **-127 to 127**) |
| unsigned char | 1 | used for characters (range **0 to 255**) |

Let's see a few examples.

long b = 4523232;

long int c = 2345342;

long double d = 233434.56343;

short d = 3434233; // Error! out of range

unsigned int a = -5; // Error! can only store positive numbers or 0

### **C++ Input & Output:**

* + 1. Output:

In C++, cout sends formatted output to standard output devices, such as the screen. We use the cout object along with the << operator for displaying output.

* + 1. Input:

In C++, cin takes formatted input from standard input devices such as the keyboard. We use the cin object along with the >> operator for taking input.