Lecture Three: C++ Programming Language

1- C++ programming language

C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a compiled language.

1. C++ is a high-level, general-purpose programming language designed for system and application programming. It was developed by Bjarne Stroustrup at Bell Labs in 1983 as an extension of the C programming language. C++ is an object-oriented, multi-paradigm language that supports procedural, functional, and generic programming styles.

2. One of the key features of C++ is its ability to support low-level, system-level programming, making it suitable for developing operating systems, device drivers, and other system software. At the same time, C++ also provides a rich set of libraries and features for high-level application programming, making it a popular choice for developing desktop applications, video games, and other complex applications.

3. C++ has a large, active community of developers and users, and a wealth of resources and tools available for learning and using the language.

4. Object-Oriented Programming: C++ supports object-oriented programming, allowing developers to create classes and objects and to define methods and properties for these objects.

5. Templates: C++ templates allow developers to write generic code that can work with any data type, making it easier to write reusable and flexible code.

6. Standard Template Library (STL): The STL provides a wide range of containers and algorithms for working with data, making it easier to write efficient and effective code.

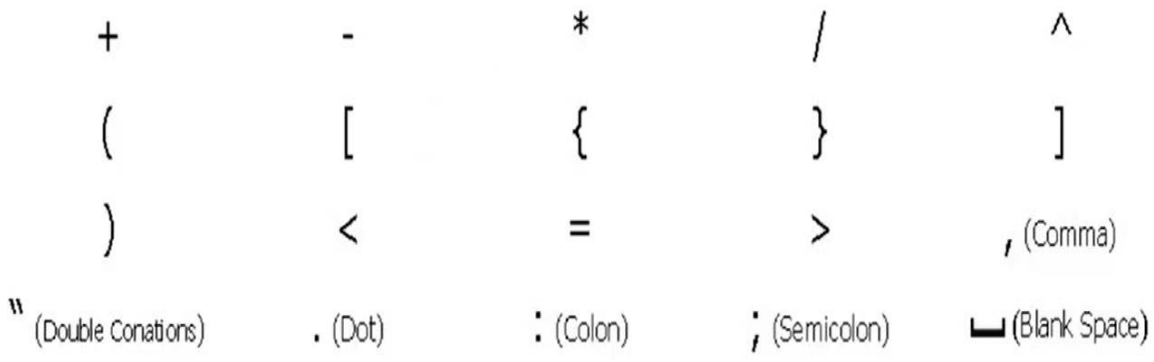
2- Character set and Identifiers

- **Characters:**

C++ has the letters and digits, as show below

- Uppercase A,B,C,….Z
- Lowercase a,b,c,…….z
- Digits 0,1,2,…..9

Special Character s: all characters other than listed treated as special characters for example



+	-	*	/	^
(	[	{	}	]
)	<	=	>	, (Comma)
" (Double Conations)	. (Dot)	: (Colon)	; (Semicolon)	␣ (Blank Space)

- **Identifiers**

An identifier is a sequence of characters used to denote one of the following:

- Object or variable name
- Class, structure, or union name
- Enumerated type name
- Member of a class, structure, union, or enumeration

3- Variable and Constant

In C++, variables can be declared anywhere in your code and used anywhere in the current block below the line where they are declared. Variables can be declared without being given a value.

- **Variable**

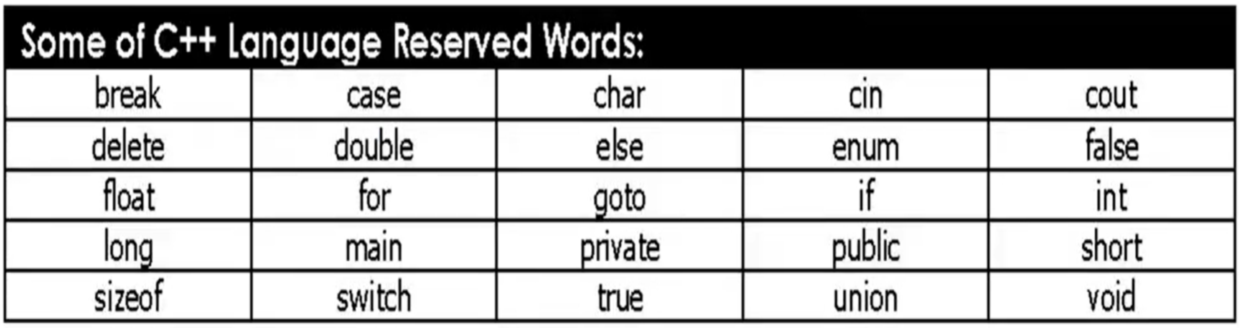
A **variable** should not begin with digits, C++ doesn't set maximum length for an identifier. **C++ allow identifiers up to 127 characters.**

For example:

1. **My_name (7 char)**
2. **i (1 char)**
3. **B (1 char)**

Keywords

The keywords are also identifiers but cannot be users identified, since they are reserved words. All the keywords should be in lower case letter. Reserved words cannot be used as variable names or constant, the following word are reserved for use as keywords



Some of C++ Language Reserved Words:				
break	case	char	cin	cout
delete	double	else	enum	false
float	for	goto	if	int
long	main	private	public	short
sizeof	switch	true	union	void

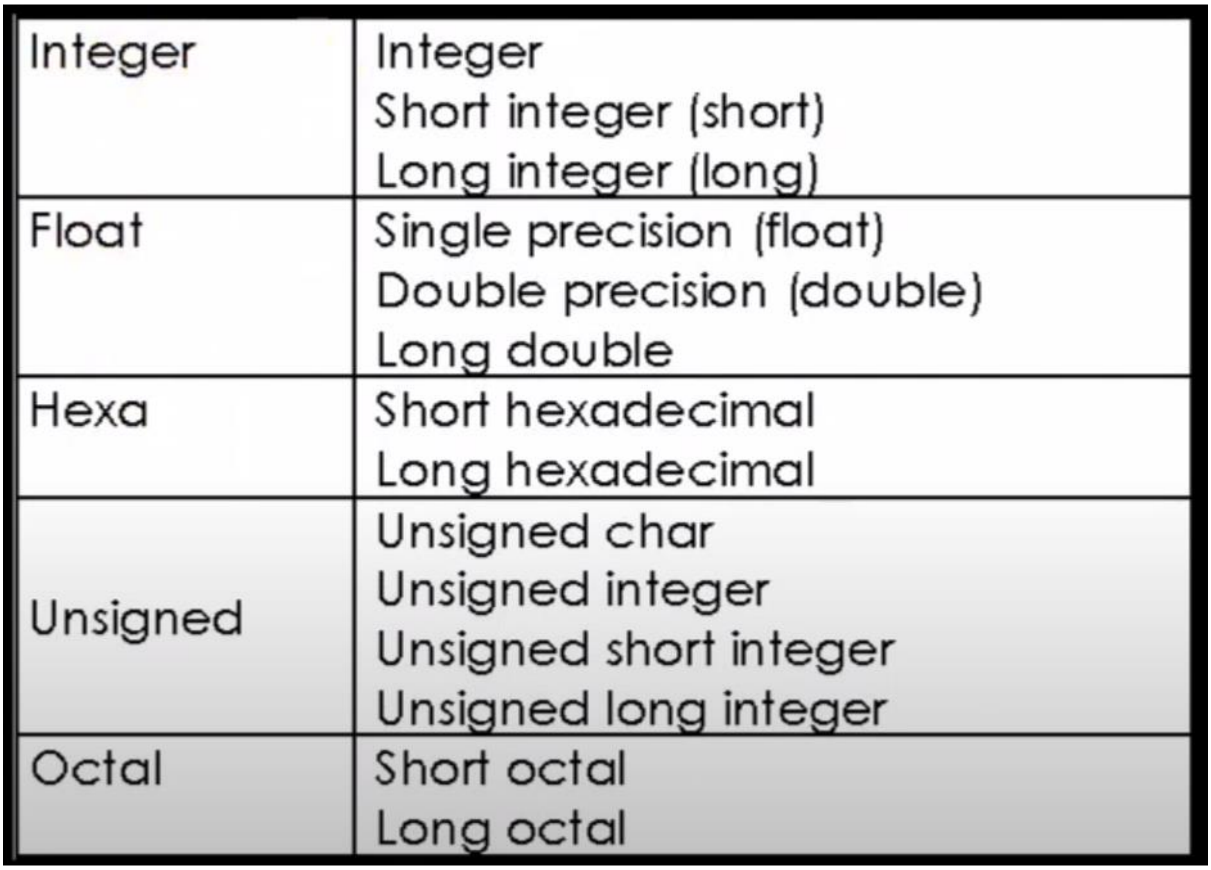
- **Constant**

There are three types of constants (Data Type):string constant, numeric constant and character constant

1-String constant: a string constant are a sequence of alphanumeric characters enclosed in double quotation marks whose maximum length is 255 characters. Like "The result =")

2-Numeric Constants: are positive or negative numbers, there are four type of numeric constants integer, floating, point and Hexadecimal

The following table shows the most common data types used in C++.



Integer	Integer Short integer (short) Long integer (long)
Float	Single precision (float) Double precision (double) Long double
Hexa	Short hexadecimal Long hexadecimal
Unsigned	Unsigned char Unsigned integer Unsigned short integer Unsigned long integer
Octal	Short octal Long octal

1-integer constants: Do not contain decimal points: int x,y shortint x,y longint x,y

2-Float point: Positive or negative number in exponential form.

3-character constant: represented within single quotes for example :'A','a',;'.

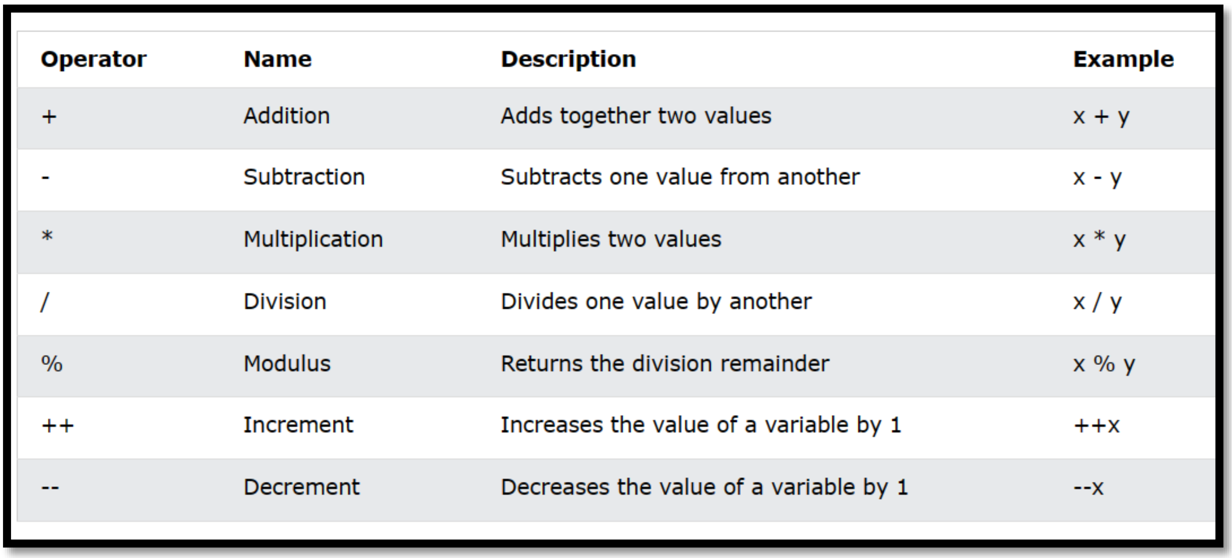
4-C++ operator

C++ divides the operators into the following groups:

- **Arithmetic operators**
- **Assignment operators**
- **Comparison operators**
- **Logical operators**
- **Bitwise operators**

A. Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.



Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x

Arithmetic operators as per precedence:

- () for grouping the variables
- -Unary for negative number
- */ multiplication & division
- + - addition and subtraction

For example:

X+Y*X-Z, where X=5,Y=6 and Z=8

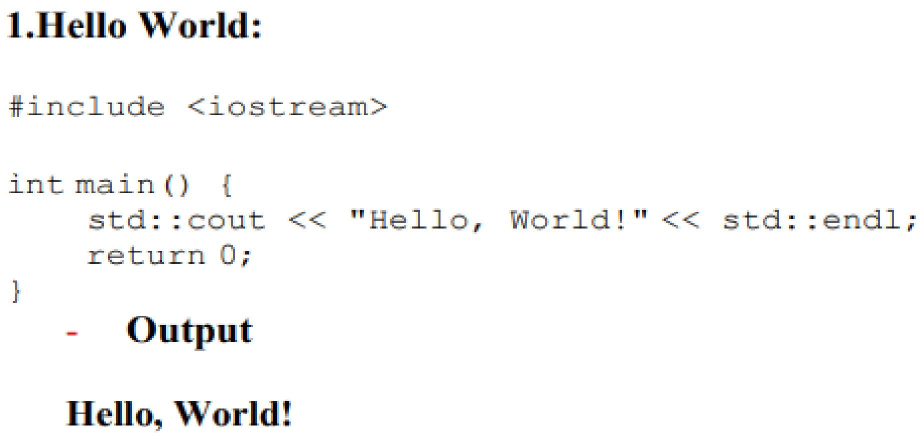
=5+(6*5)-8 >>>>> (5+30)-8 >>>>> 35-8 >>>> 27

B. Assignment Operators

Assignment operators are used to assign values to variables. In the example below, we use the **assignment** operator (=) to assign the value **10** to a variable called **x**:

int x = 10;

5- C++ Example :



1.Hello World:

```
#include <iostream>

int main() {
     std::cout << "Hello, World!" << std::endl;
     return 0;
}
```

- **Output**

Hello, World!

Steps for C++ Program Development and Execution

1. Editing
2. Compiling
3. Linking Library files
4. Loading
5. Execution