

Getting started with C++

- What is a Program?
- What do you mean by High Level Language and a Low level Language?
- What is the main difference between a compiler and an interpreter?

Tokens(Lexical Unit)

Tokens are the smallest individual unit in a program. C++ has the following tokens:

- **Keywords** :- are the words that convey a special meaning to the language compiler.

eg: int, float, if, for, while

- **Identifiers** :- They are the fundamental building blocks of a program and are used as the general terminology for the names given to different parts of the program.

Eg: Myfile, _CHK

- **Literals** :- Literals are referred to as constants, are the data items that never change their value during a program run.

C++ has several kinds of literals

1. Bool literal (true or false)
2. Integer constant (decimal, Octal and hexadecimal).
3. Character constant (escape sequence \n \t)
4. String literals (multiple character form string)
5. Floating constants(fractional and exponential part).

- **Punctuators :-**

1. Braces : { } – indicates the start of the compound statement(block of code containing more than one executable statement.
2. Brackets [] - array
3. Paranthesis – () – function call and parameters
4. Comma- , - separator in function list.
5. Semicolon - ; - statement terminator.
6. Equal sign - = - assignment , variable initialization
7. Colon - : - indicates a labeled statement.

Operators

Relational Operators

- < [less than]
- > [greater than]
- <= [less than or equal]
- >= [greater then or equal]
- == [equals]
- != [Not equals]

Always returns a True[1] or False [0] value

Logical Operators

- AND [&&] eg: X && Y
- OR [||] eg: X || Y
- NOT [!] eg: !X

Truth table for AND operator

X	Y	X && Y
T	T	T
T	F	F
F	T	F
F	F	F

Truth table for OR operator

X	Y	X Y
T	T	T
T	F	T
F	T	T
F	F	F

Truth table for NOT operator

X	!X
T	F
F	T

What is a data type?

- In computer programming, information is stored in a computer memory with different **data types**.
- We must know what is to be stored in a computer memory, whether it is a simple number, a letter or a very large number.
- As we also know, computer memory is organized in **bytes**, and for these variables with varying information a data type is associated.
- The minimum amount of memory in computer memory is a byte, that can store a small amount of data and managed easily.

- Every variable is declared with two entities, its type and its name. There are several data types available in C++.

Eg : <data type> <variable name>;

```
int a;
// int is the data type and a is the
variable;
```

- The basic built in data types or the **fundamental data types** are **char**, **int**, **float**, **double** and **void**
- C++ also allows **user defined data types** like class, structure etc.

A main() Function in C++

- ▶ C++ programs consist of modules called **functions**
- ▶ Every statement within every C++ program is contained in a function
- ▶ Every function consists of two parts:
 - A **function header** is the initial line of code in a C++ which always has three parts:
 - ▶ Return type of the function
 - ▶ Name of the function
 - ▶ Types and names of any variables enclosed in parentheses, and which the function receives
 - A **function body**

Creating a `main()` Function

- ▶ A C++ program may contain many functions, but every C++ program contains at least one function, and that function is called `main()`
- ▶ If the main function does not pass values to other programs or receives values from outside the program, then `main()` receives and returns a void type
- ▶ The body of every function in a C++ program is contained in curly braces, also known as curly brackets

Creating a `main()` Function

```
void main()  
{  
}
```

Figure 1-7 The simplest C++ program

- ▶ Every complete C++ statement ends with a semicolon
- ▶ Often several statements must be grouped together, as when several statements must occur in a loop
- ▶ In such a case, the statements have their own set of opening and closing braces within the main braces, forming a **block**

cin && cout

- cin is known as the standard input stream
- cout is known as the standard output stream
- `cout<<"hello world";`
will display **hello world** on the screen
- `int a; // declare a variable`
`a=10; // assign the variable value`
`cout<<a; // will display the value of a on the screen`
- `cout<<"a="<<a;`
`//will display a=10 on the screen`

C++ Program starts with header file
`#include<iostream.h>` ← header file
`#include<conio.h>`


```
void main()
{
    cout<<"Hello world";
    getch();
}
```

OUTPUT :
Hello world

Output operator <<

Syntax:

`cout<<" Hello ";`




Prints the text on the screen

Input operator >>

Syntax:

`cin>> var;`



Accepts an input from the user

Eg: Accept the age from the user and display it

```
int age;      // variable age is declared as integer type
cout<<"Enter your age";  // This stmt is displayed on
the monitor
cin>>age;     // value of age is inputted from the user
cout<<"The age of the person is :"<<age;
              // the age of the person is displayed
```

Output of the program is

Enter your age 56 The age of the person is:56

Variable declaration

- Syntax:

<data type> <var1>,<var2>.....;

Eg: int a,b,c;

float sum, avg;

Variable initialization

Syntax:

data type var=value;

Eg: int a=5;

float sal=250.50

What is a variable?

- A variable is a value that can change, depending on conditions or on information passed to the program.

- Eg :

- int a=5;

- a=a+10;

- cout<<"a="<<a;

- a=a*10;

- cout<<"\n A="<<a;

OUTPUT

a=15

A=150

Variables

Rules for variable names:

- Variable name should always start with an alphabet.
- Special symbols (like \$,%,-) are not permitted except underscore(_).
- Digits are permitted.
- Blank space is not permitted.
- Keywords are not permitted, but we can use the keywords along with alphabets or digits.

Egs of variable assignments

- `Sum=5;` `//valid`
- `Total sum=10 ;` `//Invalid`
- `Total-sum=10;` `//Invalid`
- `Total_sum=10;` `//valid`
- `Sum123=100;` `//valid`
- `1st Term=2400;` `//Invalid`
- `First_Term=2400;` `//valid`

Syntax in C++

- Each and every statement in c++ ends with a semicolon.

A=5; // valid

A=x+4; // valid

5=A; // Invalid

X+3=A; // Invalid

The value from the right side goes to the left side.

In c++ all the expressions and constants will come on the right side of assignment operator.

Data types in C++			
Data Type	Memory (ByteS)	Minimum Value	Maximum Value
Bool	1	Logical Value T/F	Logical Value T/F
Char / signed char	1	-128	127
Unsigned Char	1	0	255
Short	2	-32768	32767
Unsigned Short	2	0	65535
int	2	-32768	32767
unsigned int	2	0	65535
Long	4	-2147483648	2147483647
unsigned long	4	0	4294967295
float	4	10^{-38}	10^{38}
double	8	10^{-308}	10^{308}
long double	10	10^{-4932}	10^{4932}

Data type modifiers

- signed
- unsigned
- long
- short

Pgm 2

```
#include<iostream.h>
#include<conio.h>
void main()
{ clrscr(); // clears the screen
  int a;
  cout<<"Enter a number :";
  cin>>a;
  cout<<"The number is :"<<a;
  getch();
}
```

OUTPUT

Enter a number : 10 The number is:10


Initialization of variables

- Static
- Dynamic

Value Assignment

Eg: int a;

a=10; // value is assigned to a



Dynamic initialization

Meaning value is not assigned in the beginning

Eg: int sum, count;

cin>>sum,count;

float total=sum/10;

Escape sequence

- `\n` → line feed (new line)
- `\t` → tab space (8 spaces)
- `\b` → back space
- `\a` → alert sound (beep)

Q. Write the stmt to get the foll output.

1. BGS	2. B	3. BGS	NPS
NPS	G		
	S		
	N		
	P		
	S		

Eg: Accept the age from the user and display it

```
Int age;
cout<<"Enter your age : \n";
cin>>age;
cout<<"\n The age of the person is :\t "<<age;
```

Output of the program is

```
Enter your age
56
The age of the person is:      56
```


Arithmetic operators

Operators	Usage	Examples
+ Sum	$x+1$ or $x+y$	$X=5, y=6$ $\text{Sum} = X+y$ $\text{Sum}=11$
- Difference	$x-1$ or $x-y$	$\text{Diff}=x-y$ $\text{Diff}=-1$
* Product	$x*3$ or $x*y$	$\text{Prod}=x*y$ $\text{Prod}=30$
/ Quotient	$c/3$ or x/y	$X=5, y=2$ $\text{Quot}=X/y$ $\text{Quot}=2.5$
% Remainder	$s\%2$ or $x\%y$	$\text{Rem}=x\%y$ $\text{Rem}=1$

W.A.P to enter the user class and section and display then.

```
#include<iostream.h>
#include<conio.h>
void main()
{ clrscr();
  int cl; char sec;
  cout<<" Enter the person's class and section :";
  cin>>cl; cin>>sec
  cout<<" \n the person's class is : " <<cl<<" section : " <<sec;
  getch();
}
```

Output

Enter the person's class and section : 5 

A

The person's class is: 5 section A

2. W.A.P to enter the user age and display his age after 15 years.

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int age,nage;
    cout<<" Enter the person's age :";
    cin>> age;
    nage=age+15 // or age=age+15
    cout<<" \n the person's age after 15 years :" <<nage; // or age
    getch();
}
```

Output

Enter the person's age: 23 
the person's age after 15 years : 38

3. W.A.P to accept your marks in three subject and display them

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
void main()
{
    clrscr();
    int m1,m2,m3;
    cout<<" Enter Science mark:";
    cin>> m1;
    cout<<" \nEnter Maths mark:";
    cin>> m2;
    cout<<"\n Enter English mark:";
    cin>> m3;
    cout<<" \n Science marks :" <<m1<<"\n Maths marks :"<<m2<<" \n English
    marks :"<<m3;
    getch();
}
```

Escape sequence

Q. Write the pgm to get the following output.

4. BGS
NPS

5. B
G
S
N
P
S

6. BGS NPS

7. To find the circumference and area of a circle
8. To find the sum, product and difference of any two given number.
9. To find the percentage of the student in 5 main subject
10. To find the area of a triangle.
11. Volume of a sphere
12. Volume of cylinder
13. Volume of Cube.