

Important Points

- In C++ sequence of bytes corresponding to input and output are commonly known as streams. There are two types of streams:
 - Input stream** - In the input stream, the direction of the flow of bytes occurs from the input device (for ex keyboard) to the main memory.
 - Output stream** - In output stream, the direction of flow of bytes occurs from main memory to the output device (for ex-display)The sign "<<" is called insertion operator
The sign ">>" is called extraction operator
- To access the value of the global variable "c," we use **scope resolution operator** "::" with the "c" variable, if local variable "c" exist in the function.
- double is the default type given to a decimal literal (34.4 is double by default and not float), so to use it as float, you have to specify it like "34.4F,"
- In C++, there are two ways to typecast a variable, either using "(float)a" or using "float(a)"
- C++ forbids converting a string constant to char*
- language manipulators are used in the formatting of output.
 - "endl" is used for the next line.
 - "setw" is used to specify the width of the output.
 - Spaces are added.
- To use ampersand "&" with the array name for assigning the address to a pointer is wrong.

Types of Variables in C++

`int x=5;` →ordinary variable

`int *p;` →pointer variable

`p = &X;`

`int &y = x;` →reference variable

`y++;` → This statement will increase the value of x by 1.

Reference Variable:-

- Reference means address.
- Reference variable is an internal pointer.
- Declaration of reference variable is preceded with ‘&’ symbol (but do not read it as ‘address of’)
- Reference variable must be initialized during declaration.
- It can be initialized with already declared variables only.
- Pointer variables can be updated as many no. of times we want but reference variables, once declared, cannot be updated with another ordinal variable.

Call by Value In case of call by value the copies of actual parameters are sent to the formal parameter, which means that if we change the values inside the function that will not affect the actual values.

Call by Pointer In the case of call by pointer, the address of actual parameters is sent to the formal parameter, which means that if we change the values inside the function that will affect the actual values.

Call by Reference In the case of call by reference, the reference of actual parameters is sent to the formal parameter, which means that if we change the values inside the function that will affect the actual values.

→ In case of call by ref. we don't pass the address of actual arguments, it is automatically passed whereas in case of call by pointer we need to pass the address of actual arguments.

Procedure Oriented

