# <u>C++</u>

C++ is an object oriented programming language. C++ was developed by Bjarne Stroustrup at AT & T labouratories in USA in the early eighties.

Strousstrup, an admires of Simula 67 and a strong supporter of C, wanted to combine the best of both languages and create a more powerful language that could support object oriented programming features and still retain the power and elegance of C. The result was C++.

Therefore, C++ is an extension of C with a major addition of the class construct feature of Simula 67. Sometimes, it is called "C with classes". Later in 1983, the name was changed to C++. The idea of C++ comes from the C increment operator ++,

i.e. C++ is an augmented (incremented) version of C.

C++ is a superset of C. Almost every correct statement in C is also a correct statement in C++, although the reverse is not true.

The three most important facilities that C++ adds on to C are classes, function overloading, operator overloading.

The object oriented features in C++ allow programmers to build large programs with clarity, extensibility and ease of maintenance, incorporating the spirit and efficiency of C.

# Application of C++

We can use C++ for any programming task including development of editors, compilers, databases, communication systems and any complex real life application system.

### Example:



So, we must include appropriate header files depending on the contents of the program. Say if we want to use printf() and scanf() functions, the header file "stdio.h" must be included.



The operator << is also known as bitwise left shift operator and it can still be used for this purpose. This is an example of how one operator can be used for different purposes, depending on the context. This concept is known as operator overloading, an important aspect of polymorphism.

# **Comments:**

C++ introduces a new comment symbol (//)

// C++ program
// app. of C++
// an example
or /\* an example
// app.of C++ \*/
multiline comment

# **Return statement**

In C++, main() returns an integer type value to the operating system. [Where as in C, main() does not return any value and no return statement].

Therefore, in C++, main() should be like that



Most C++ compilers will generate an error or warning if there is no return statement.

May operating systems test the return value (or exit value) to determine. If there is any problem. An exit value of zero means the program ran successfully, while a non-zero value means there was a problem.

# Example



>>  $\rightarrow$  Operator extracts (or takes) the value from the keyboard and assigns it to the variable on its right.



In this statement cout <<"Sum" << Sum << "\n";

We have used the extraction operator repeatedly. The multiple use of << in one statement is called Cascading. We can also cascade input operator >> as shown,

# cin >> num1 >> num2;

If we key in two values, say 10 and 20 then 10 will be assigned to num1 and 20 to num2.

### Structure of C++ Program.

A typical C++ program would contain four sections as show below.

Include file
Class Declaration
Class Function Definition
Main Function Program

We use .cpp to indicate that this file is a C++ file in Turbo C++ and Borland C++.

This is source program Save it to the disk by selecting Save from the file menu or by pressing  $f_2$ .

### **Compiling and Linking**

A source file (an ASCII file similar to that generated by a word processing is not an executable program; it is only the instructions on how to create a program.

#### Transforming your source file into an executable program requires two steps.

(1) **Compile:** the source file into an object file which has •OBJ extension, contains m/c language instructions that can be executed by the computer.

(2) **Linking is required**: The second step is necessary because an executable program almost always consists of more than one object file. Linking combines the object files into a single executable program.

#### Why does a program consists of more than one object file.

(a) The programmer may have divided the program into several source files. Each of these source files is then compiled into a separate object file and these object files must be linked together.

(b) Library functions always come in object file form and must be combined with the userwritten program.

 $ctrl + f_9$ - Run the program

Alt  $+ f_5 - to see o/p screen$ .

#### Errors

**Complier Error:** Suppose you have forgotten to type the semicolon ; at the end of statement then it will generate error.

Press any key , a message window will appear. Press  ${\rm f}_6$  to move from the message window to the edit window.

Linear error: Suppose you spell the word

main() as maid()

Then program will compile successfully but have linker error. The linker must find a f<sup>n</sup> main, without this, it cannot create an executable file.

**Run-time error:** don't reveal themselves until the program executes. These errors include divide by 0, stack overflow etc.

## **Example with Class:**

```
# include <iostream.h>
                           They provide a method of binding together data &
class person
                             functions which operate on them. Like structure in C,
                             classes are user-defined data types.
{
       char name [30];
       int age;
       public:
           void getdata (void);
           void display (void);
};
void person : : getdata (void)
{
       cout << "Enter name";
       cin >> name;
       cout << "Enter age";
       cin >> age;
}
void person : : display (void)
{
       cout << "\n name" << name;
       cout << "\n Age:"<< age;
}
main()
{
        person p;
        p.get data();
        p.display();
```

The program defines person is a class. It includes two basic data type items and two functions to operate on that data. The functions are called member functions.

p is an object of type person. Class objects are used to invoke functions defined in that class.