M.Sc. III Sem. (Mathematics)

Paper 2nd - Fundamentals of Computer Science - I

Unit II

Reference Book : **E. Balagurusamy, *Object Oriented Programming with C++*, McGraw Hill Education (India) Pvt. Ltd., New Delhi.**

Topic : Beginning with C++

WHAT IS C++?

C++ is an object-oriented programming language. It was developed by Bjarne Stroustrup at AT&T Bell Laboratories in Murray Hill, New Jersey, USA, in the early 1980's. Stroustrup, an admirer of Simula67 and a strong supporter of C, wanted to combine the best of both the 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 Simula67. Since the class was a major addition to the original C language, Stroustrup initially called the new language 'C with classes'. However, later in 1983, the name was changed to C++. The idea of C++ comes from the C increment operator ++, thereby suggesting that C++ is an augmented (incremented) version of C.

C++ is a superset of C. The most important facilities that C++ adds on to C are classes, inheritance, function overloading, and operator overloading. These features enable creation of abstract data types, inherit properties from existing data types and support polymorphism, thereby making C++ a truly object-oriented language.

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. The addition of new features has transformed C from a language that currently facilitates top-down, structured design, to one that provides bottom-up, object-oriented design.

A SIMPLE C++PROGRAM

Let us begin with a simple example of a C++ program that prints a string on the screen.

Program for printing a string.

```
#include <iostream>        // include header file
using namespace std;
int main()
{
        cout << "C++ is better than C.\n";  //  C+ + statement
        return 0;
}       // End of example
```

This simple program demonstrates several C++ features.

Program Features :

Like C, the C++ program is a collection of functions. The above example contains only one function, main( ). As usual, execution begins at main( ). Every C++ program must have a main( ). C++ is a free-form language. With a few exceptions, the compiler ignores carriage returns and white spaces. Like C, the C++ statements terminate with semicolons.

Comments :

C++ introduces a new comment symbol // (double slash). Comments start with a double slash symbol and terminate at the end of the line. A comment may start anywhere in the line, and whatever follows till the end of the line is ignored. Note that there is no closing symbol.

The double slash comment is basically a single line comment. Multiline comments can be written as follows:

```
//	This  is an example of
//	C+ + program  to illustrate
//	Some  of its  features
```

Output Operator :

The statement

```
	cout  <<  "C++  is  better  than  C.";
```

causes the string in quotation marks to be displayed on the screen. This statement introduces two new C++ features, cout and <<.

The identifier cout (pronounced as 'C out') is a predefined object that represents the standard output stream in C++. Here, the standard output stream represents the screen. It is also possible to redirect the output to other output devices. We shall later discuss streams in detail.

The operator **<<** is called the **insertion** or **put to** operator. It inserts (or sends) the contents of the variable on its right to the object on its left.

You may recall that the operator << is the bit-wise 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.

Input Operator :

The statement

```
	cin  >>  num1;
```

is an input statement and causes the program to wait for the user to type in a number. The number keyed in is placed in the variable numl. The identifier cin (pronounced 'C in') is a predefined object in C++ that corresponds to the standard input stream. Here, this stream represents the keyboard.

The operator **>>** is known as **extraction** or **get from** operator. It extracts (or takes) the value from the keyboard and assigns it to the variable on its right. Like <<, the operator >> can also be overloaded.

The iostream file :

A C++ program typically contains pre-processor directive statements at the beginning. Such statements are preceded with a # symbol to indicate the presence of a pre-processor directive to the compiler. All C++ programs begin with a #include directive that includes the specified header file contents into the main program.

We have used the following #include directive in the program:

```
#include  <iostream>
```

This directive causes the preprocessor to add the contents of the iostream file to the program. It contains declarations for the identifier cout, cin and the operator <<, >>. Some old versions of C++ use a header file called iostream.h. This is one of the changes introduced by ANSI C++. (We should use iostream.h if the compiler does not support ANSI C++ features.)

The header file iostream should be included at the beginning of all programs that use input/output statements.

Namespace :

Namespace is a new concept introduced by the ANSI C++ standards committee. This defines a scope for the identifiers that are used in a program. For using the identifiers defined in the namespace scope we must include the using directive, like

```
using namespace  std;
```

Here, std is the namespace where ANSI C++ standard class libraries are defined. All ANSI C++ programs must include this directive. This will bring all the identifiers defined in std to the current global scope, using and namespace are the new keywords of C++.