

Features of C++

Some Important features of “C++ are as follow:-

1. Convenient Language

C++ is very convenient language. It provides many facilities, in easier way that are difficult to use in low level languages .Programmers can write complex programs more easily as compared to low-level languages.

2. Well-Structured Language

C++ is a Well-Structured language. Its syntax is very easy to understand. The programs written in C++ language are easy to maintain and modify.

3. Case Sensitivity

C++ is case sensitive language. It means that it can differentiate uppercase and lowercase. words. All keywords are written in lowercase. This feature makes it easier to maintain the source code.

4.Machine Independence

C++ language provides machine independence. It means that the programs written in C++ language can be executed on different types of computers. For example, program written in C + + can be executed on Intel processors and Motorola processors with a little modification .It is preferable to write program in C++ rather than machine language.

5. Object Oriented

C++ is an object oriented language. in object oriented technique, the programs are written on the basis of objects. An object is a collection of data and functions. Object may represent a person, thing or place in real world. In OOP, data and all possible functions on data are grouped together. Object oriented programs are easier to learn and modify.

6. Modular Programming

C++ language provides the facility of modular programming. It means that a program can be divided into small modules. These modules can be developed and compiled independently and then linked together.

7. Standard Libraries

The standard C++ library is a set of functions, constants, classes and objects to extend C++ language. It provides the basic functionality to interact with the operating system. These libraries can be reused by any programmer for writing programs more easily in less time.

8. Hardware Control

C++ language provides close control on hardware. The programmer can write efficient programs to control hardware components of computer system.

9. Brevity

C++ is a small language. It has a small number of keywords and programming controls. But still it is every powerful for developing different types of programs. The code written in C++ is very short as compared with other languages.

10.Speed

C++ compilers generate very fast code. The resulting code from a C++ compilation is very efficient. This code executes very efficiently. So the programs take less time to execute.

11. Popular Language

C++ is a very widely used programming language. There are many tools available for C++ programming and a broad base of programmers contributing to C++ community.

12. Compatibility

C++ is backward compatible with C language. Any code written in C, can easily be included in a C++ program without hardly making any change.

Basic Structure Of C++ Program

The format of writing program in C++ is called Its structure. The basic structure of C++ program is very flexible. It increases the power of language. It consists of the following parts:

- Preprocessor directive.
- Main() function.
- Program body (C++ statements).

Preprocessor Directive

Preprocessor directive is an instruction given to the compiler before the execution of actual program. Preprocessor directive is also known as compiler directive. The preprocessor directives are processed by a program known as Preprocessor. It is part of C++ compiler .it modifies C++ source program before compilation. The semicolon is not used at the end of preprocessor directives.

The preprocessor directives start with hash symbol #. These directives are written at the start of program. The following preprocessor directive is used in C++ to include header files in the program.

Include preprocessor

Include preprocessor directive is used to include header files in the program. The syntax of using this directive is as follows:

```
#include <iostream .h>
```

The above statement tells the compiler to include the file **iostream.h** in source program before compiling it.

Header File.

Header files are collection of standard library functions to perform different tasks .There are many header files for different purposes. Each header file contains different types predefined functions. Many header files can be included in one program. The header file must be Included in the program before calling any of its functions in the program.

The extension of a header file is **.h**. The **include** preprocessor directive is used to include header files in programs. These files are provided by C++ compiler system.

The header files are normally stored in **INCLUDE** subdirectory. The name of header file is written in angle brackets.

Syntax:-

The syntax of using header files is as follows:

```
#include <header file name>
```

The name of header file can also be used in double quotes as follow:

```
#include "header_file_name".
```

Example:-

```
#include "iostream.h"
```

The word "iostream" stands for **input/output stream**. This Header file contains the definitions of built-in input and output functions and Objects.

A header file **math.h** is used in programs to use predefined mathematical functions.

The following statement is used to include this file in program:

```
#include "math.h"
```

main() Function

The **main()** function is the starting point of a C++ program. When the program is run, the control enters **main()** function and starts executing its statements.

Each program must contain **main()** function. If a program does not contain main function, it can be compiled but cannot be executed. Any number of statements can be written in the body of the **main()** function.

The syntax of **main()** function is as follows:

```
void main( )  
{  
  body of main function  
}
```

C++ Statements

The statements of the program are written in curly braces. The curly brace { is called **opening brace** and) is called **closing brace**. The braces are also known as **delimiters**. These statements are collectively known as the **body** of a program. Each statement in C++ language is terminated by **semicolon(;**).

A statement in C++ language is an instruction for the computer to perform a task. Computer performs these instructions one by one in the same sequence in which these instructions are written.

Example

The following example explains the basic structure of C++ program:

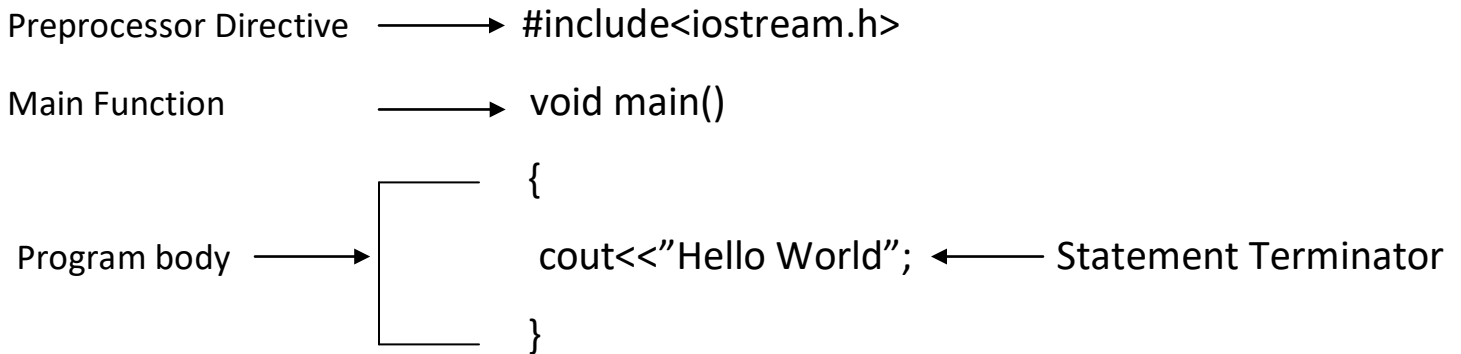


Figure: Basic Structure of C++ Program

In the above example,

- The first line is preprocessor directive to include a header file `iostream.h`.
- The second line is main function where the execution of program starts.

Using 'cout'

The **cout** object is used to print a message on the screen. The message may consist of strings and values. A **string** is a set of characters. The **cout** object is used with the insertion operator `<<`. Anything written after the insertion operator is displayed on the screen.

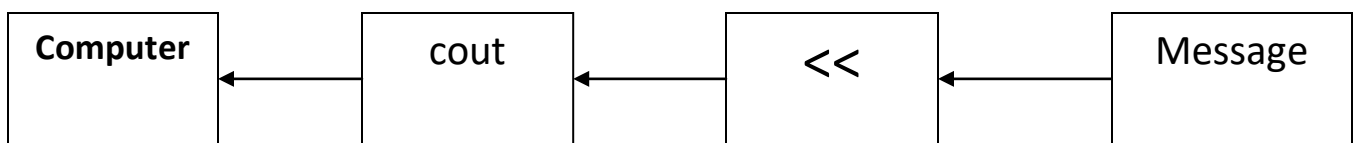


Figure: Working of 'cout' object

Token

A **token** is a language element that is used to form a statement. C++ statement may consist of different tokens. Different types of tokens are as follows:

- **Keywords:** Keyword is a word in C language that has a predefined meaning and purpose. The meaning and purpose of a keyword is defined by the developer of the language.

Example: double for If const

- **Identifiers:** Identifier is the name of a variable or function etc. It is also called **user-defined word**.

Example: Student age number_of_character item20 counter

- **Constants:** Constant is a quantity that cannot be changed during execution of a program.

Example: -220.7 'j' 420.7 213 520

- **String literals:** A collection of characters written in double quotations is called string or string literal.

Example: "This is a string constant" "120" "99-Mall Road, Lahore"

- **Operators:** Operator is a symbol that performs some operation. It acts on different operands.

Example: + - * / %

- **Punctuators:** Punctuator is a symbol that is used to separate two tokens.

Example: [] () {} , ; . :

White Spaces

The white spaces are used in programs to increase readability. Different types of white spaces include space, tab and carriage return etc. C++ compiler ignores the white spaces. All statements of a program can be written on one line but it will be difficult to read. The white spaces make the source program more readable. A single statement can be written on two or more lines or order to increase readability but it will remain same for the compiler.

Identifier

The identifiers are the names used to represent variable, constants, types, functions and labels in the program. Identifier is an important feature of all computer languages. A good identifier name should be descriptive but short.

An identifier in C++ may consist of 31 characters. If the name of an identifier is longer than 31 characters, the first 31 character will be used. The remaining characters will be ignored by C++ compiler. Some important rules for identifier name are as follows

- The first character must be an alphabetic or underscore (_).
- The identifier name must consist of only alphabetic characters, digits or underscores.
- The reserved word cannot be used as identifier name

Types of Identifiers

C++ provides the following types of identifiers:

1. Standard Identifiers

A type of identifier that has special meaning in C++ is known as **standard identifier**. C++ cannot use a standard identifier for its original purpose if it is redefined.

Example

cout and **cin** are examples of standard identifiers. These are the names of input/output objects defined in standard input/output library **iostream.h**.

2. User-defined Identifiers

The type of identifier that is defined by the programmer to access memory location is known as **user-defined identifier**. The user-defined identifiers are used to store data and program results.

Example

Some examples of user-defined identifiers are **a**, **marks** and **age** etc.

Keywords

Keyword is a word in C++ language that has a predefined meaning and purpose. The meaning and purpose of a keyword is defined by the developer of the language. It cannot be changed or redefined by the user. Keyword can be used for the same purpose for which it is defined, Keywords are also known as **reserved words**. There are different types of keywords in C++ language. The total number of keywords is 63.

List of Keywords

asm	auto	bool	break	case	catch	char	class
const	const_cast	continue	default	delete	do	double	
dynamic_cast	else	enum	explicit	export	extern		
false	float	friend	goto	if	inline	int	mutable
namespace	new	operator	private	protected	public		

register reinterpret_cast return short signed sizeof
static static_cast struct switch template this
throw true try typedef typeid typename union
unsigned using virtual void volatile wchar_t
while

Data Types

The data type defines a set of values and a set of operations on those values. The computer manipulates various types of data. The data is given to the program as input. The data is processed according to the program instructions and output is returned. The data and its type are defined before designing the actual program used to process the data. The type of each data value is identified at the beginning of program design.

A C++ program may need to process different types of data. Each data type require different amount of memory. C++ language provides the following data types:

Data type	Purpose
int	To store numeric values
float	To store real values
double	To store large real values
Char	To store character values

Table: Data types in C++ language

Integer Data Types

Integer data is numeric value with no decimal point or fraction. It includes both positive and negative values. The minus sign - is used to indicate negative value. If no sign is used, the value is positive by default.

Examples

Some examples of integer values are 10, 520 and -20 etc.

Types of Integers

C++ provides different types of integer data. These are as follows:

Data Type	Size in Bytes	Description
int	2	Ranges from -32,768 to 32,767.
short	2	Ranges from -32,768 to 32,767.
unsigned int	2	Ranges from 0 to 65,535.
long	4	Ranges from -2,147,483,648 to 2,147,483,647.
unsigned long	4	Ranges from 0 to 4,294,967,295.

Table: Data types for integers

1. int Data Type

int data type is used to store integer values. It takes two or four bytes in memory depending on the computer and compiler being used. In **MS-DOS**, it takes two bytes and its range is from -32768 to 32767.

INT_MIN represents the smallest value that can be stored in **int** data type. **INT_MAX** represents the largest value that can be stored in **int** data type. Both **INT_MIN** and **INT_MAX** are found in **limits.h** header file.

2. short int Data Type

short int data type is used to store integer values. It takes two bytes in memory. Its range is from -32768 to 32767.

SHRT_MIN represents the smallest value that can be stored in **short int** data type. **SHRT_MAX** represents the largest value that can be stored **short int** data type.

3. unsigned int Data Type

unsigned int data type is used to store only positive integer values. It takes two bytes in memory. Its range is from 0 to 65,535.

UINT_MAX represents the largest value that can be stored in **unsigned int** data type. It is found in **limits.h** header file.

4. long int Data Type

long int data type is used to store large integer values. It takes four bytes in memory. Its range is from -2,147,483,648 to 2,147,483,647.

LONG_MIN represents the smallest value that can be stored in **long int** data type.

LONG_MAX represents the largest value that can be stored in **long int** data type. Both

LONG_MIN and **LONG_MAX** are found in **limits.h** header file.

5. unsigned long int Data Type

unsigned long int data type is used to store large positive integer values. It takes four bytes in memory. Its range is from 0 to 4,294,967,295.

ULONG_MAX represents the largest value that can be stored in **unsigned long int** data type. It is found in **limits.h** header file.

Real Data Types

Real data is numeric value with decimal point or fraction. It is also called floating point number. It includes both positive and negative values. The minus sign - is used to indicate negative value. If no sign is used, the value is positive by default.

Examples

Some examples of real values are 10.5, 5.3 and -10.91 etc.

Types of Real

C++ provides different types of real data. These are as follows:

Data Type	Size in Bytes	Description
float	4	3.4×10^{-38} to $3.4 \times 10^{+38}$
double	8	1.7×10^{-308} to $1.7 \times 10^{+308}$
long double	10	1.7×10^{-4932} to $1.7 \times 10^{+4932}$

Table: Data types for real

1. float Data Type

float data type is used to store real values. It takes four bytes in memory. Its range is from 3.4×10^{-38} to $3.4 \times 10^{+38}$. It provides accuracy of 6 decimal places. **FLT_MIN** represents the smallest value that can be stored in **float** data type **FLT_MAX** represents the largest value that can be stored in **float** data type. Both **FLT_MIN** and **FLT_MAX** are found in **float.h** header file.

2. double Data Type

double data type is used to store large real values. It takes eight bytes in memory. Its range is from 1.7×10^{-308} to $1.7 \times 10^{+308}$. It provides accuracy of 15 decimal places.

DBL_MIN represents the smallest value that can be stored in **double** data type. **DBL_MAX** represents the largest value that can be stored in **double** data type. Both **DBL_MIN** and **DBL_MAX** are found in **float.h** header file.

3. long double Data Type

long double data type is used to store very large real values. It takes ten bytes in memory. Its range is from 1.7×10^{-4932} to $1.7 \times 10^{+4932}$. It provides accuracy of 19 decimal places.

LDBL_MIN represents the smallest value that can be stored in **long double** data type. **LDBL_MAX** represents the largest value that can be stored in **long double data** type. Both **LDBL_MIN** and **LDBL_MAX** are found in **float.h** header file.

Character Data Type

char data type is used to store character value. It takes 1 byte in memory. It is used to represent a letter, number or punctuation mark and a few other symbols.

Character values are normally given in single quotes. It can represent individual characters such as 'a', 'x', '5', and #. The character '5' is manipulated differently than integer 5. It is possible to perform mathematical operation on character values. The characters can be added, subtracted and compared like numbers.

Example

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    char ch1,ch2,sum;
    ch1 = 'A';
    ch2 = 'B';
    cout<<"Characters are :"<<ch1<<ch2;
    getch();
}
```

Output:

Characters are: AB

CHAR_MIN represents the smallest value that can be stored in **char** data type.

CHAR_MAX represents the largest value that can be stored in **char** data type. Both **CHAR_MIN** and **CHAR_MAX** are found in **limits.h** header file.

Integer Overflow and Underflow

Integer overflow occurs when the value assigned to an integer variable is more than maximum possible value. Integer underflow occurs when the value assigned to an integer variable is less than possible minimum value.

An integer variable can store values from -32768 to 32767. If the assigned value is more than 32767, it is known as **integer overflow**. If the assigned value is less than -32768, it is called **integer underflow**.

Integer overflow and underflow is not detected by the compiler. But the results may become wrong. It is the responsibility of the user to ensure that the values assigned to integer variables are within the allowed range.

Question:

Write a program that explains the concept of overflow and underflow.