Handbook on C++ Language

HANDBOOK ON C++ LANGUAGE

By

Sangita Sardana

Lecturer, Dept. of Computer Science

DAV School, Chandigarh

FIREWAL MEDIA

(An Imprint of Laxmi Publications Pvt. Ltd.)

BANGALORE • CHENNAI • COCHIN • GUWAHATI • HYDERABAD
JALANDHAR • KOLKATA • LUCKNOW • MUMBAI • PATNA
RANCHI • NEW DELHI

Published by :

FIREWAL MEDIA

(*An Imprint of Laxmi Publications Pvt. Ltd.*)
113, Golden House, Daryaganj,
New Delhi-110002

Phone : 011-43 53 25 00
Fax : 011-43 53 25 28

www.laxmipublications.com
info@laxmipublications.com

Price: **Rs. 175.00** *Only.* *First Edition*: 2011

OFFICES

Bangalore	080-26 61 15 61	**Chennai**	044-24 34 47 26
Cochin	0484-237 70 04, 405 13 03	**Guwahati**	0361-254 36 69, 251 38 81
Hyderabad	040-24 65 23 33	**Jalandhar**	0181-222 12 72
Kolkata	033-22 27 43 84	**Lucknow**	0522-220 95 78
Mumbai	022-24 91 54 15, 24 92 78 69	**Patna**	09893476827
Ranchi	0651-221 47 64		

FC+-3080-175-HANDBOOK ON C++ LANG-SAR **C—**

Typesetted at : Shubham Composer, N. Delhi. *Printed at :* LB Enterprises, Delhi.

CONTENTS

PREFACE

I am highly glad to present a series on Computer Languages. Writing a book to serve students community, gives a great feeling of satisfaction. Various books on High Level Languages are available in the market but it has been my desire to present such a series on computer languages which could serve the students community in some different manner which could be well acceptable by all. Generally students feel that the books available on the high do not cater to their requirements and there is a dearth of programming examples and graded programs. With the inspiration provided by my students, I have been able to frame a series of four different computer languages C, C++, Java and Visual Basic. Presentation in all the four books is almost on the similar lines.

This treatise covers chapters in uniform manner *i.e.*, from beginning level to advanced level of C ++ language. Advanced topics like Files, Linked Lists, Stacks, queues etc. have been explained in an understandable manner. Each and every topic has been explained with examples and illustrations.

A sufficient number of programming examples has been provided in the book after the topics and as a separate portion in the form of "More Worked Out Examples". Actual screen shots have been given both with program listings and output run.

All efforts have been made to provide the text in an error-free form yet there is a chance within human limitations that some error might have crept in inadvertently. Readers are requested to brought such errors in our notice so that future edition of this book could be improved. The author can be contacted at sangitsardana@yahoo.co.in.

—Author

1

INTRODUCTION TO C++

INTRODUCTION

C++ is a powerful programming language which has been derived from C language. C++ high level retains most of its C heritage. It has adopted basic data types, operations and program structure from C language. C++ was developed in 80's by Bjarne Stroustrup. Simply we can say that C++ is a superset of C language with extensions and improvements and object oriented features included in it.

CHARACTERISTICS OF C++

C++ is an object oriented programming language with a huge library of functions. Its popularity is increasing rapidly and programmers are inclining towards C++ due to its multiple features.

1. C++ is highly flexible language with versatility. It can be used for developing system software viz., operating systems, compilers, editors and data bases.
2. C++ is ideally suited for development of reusable software. This helps in cost reduction of software development.
3. There is no need of re-written software at the time when a computer is upgraded. C++ is a machine independent language.

CHARACTERS USED IN C++

The set of characters used in C++ consists of alphabets, digits and special characters.

The following table represents the characters used in this language.

Letters	A B C D E F G ... XYZ
	a b c d e f g...xyz
Numeric Digits	0 1 2 3 4 5 6 7 8 9

Special Characters :

(*i*)	Arithmetic Operators	–, +, *, /, %
(*ii*)	Logical Operators	&, !
(*iii*)	Brackets	( ) { } [ ]
(*iv*)	Relational Operators	< > = #
(*v*)	Other Symbols	:;_ (underscore) > > ?

C++ TOKENS

The group of characters which logically belong together is called Token. Tokens are of following four types.

1. Identifiers
2. Keywords
3. Constants
4. Operators

Identifiers

A program can use various data items with symbolic names in C++. Identifiers refer to the names of functions, arrays, variables, classes etc., created by a programmer. There are certain rules for naming these indentifiers in each language. An identifier, generally, contains symbolic name. These symbolic names are used in various C++ programs. The rules of naming identifiers in C and C++ are similar.

Rules for Naming Identifiers

1. Uppercase letters and lower case letters are distinct. There is a difference between Delhi and DELHI.
2. The name of identifier cannot begin with a digit. 1 DELHI is an invalid name. However, the first character can be selected as an underscore(_).
3. Only alphabetic characters, digits and underscore(_) are permitted in this language.
4. Other special characters are not allowed for naming a variable. White space is also not allowed.
5. The declared key word cannot be used as a variable name.

It is essential to declare variables before their usage in the program. A declaration statement lists the type and number of the variable.

For example

int *i* = 0;	Here i is declared as integer and has been initialized with a value of 0.
char father_name;	Here father_name is character variable.
float mth_sale;	Here mth_sale is a floating point variable.

Keywords

In C++, keywords are reserved identifiers which cannot be used as names for the variables in a program. C++ reserves 63 words for its own use. However the given 48 keywords are mostly and as keywords. The set of C++ keywords has been given in the following list.

asm	double	new	switch
auto	else	operator	template
break	enum	private	this
case	extern	protected	throw
catch	float	public	try
char	for	register	typedef
class	friend	return	union
const	goto	short	unsigned
continue	if	signed	virtual
default	inline	sizeof	void
delete	int	static	volatile
do	long	struct	while

Constants

The items whose values are not changed during the execution of the program are called constants. If an attempt is made to change their values, the program will not successfully execute and generate errors. Each constant is determined by its value and form. C++ has three types of constants.

(*i*) Numeric Constant (*ii*) Character Constant

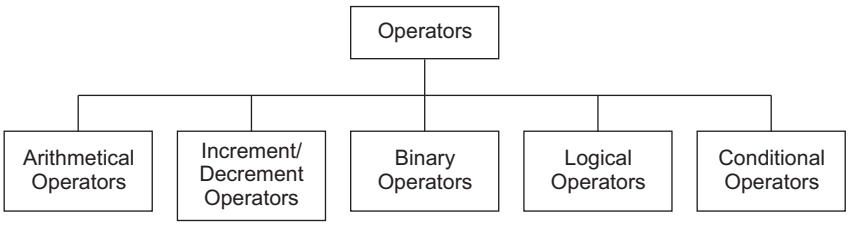
(*iii*) String Constant

Operators

An operator is a symbol or letter which causes the compiler to take an action and yield a value. An operator acts on different data items / entities called **Operands**.

For example

$x + y$ is an expression in which sign + is an operator which specifies the operation of addition.



Arithmetical Operators

Arithmetical operators are used to perform various mathematical calculations. The result of an arithmetical expression is a numerical value. An arithmetical expression may be a variable or a constant or a combination of both. These variables or functions are connected with arithmetical operators.

Increment and Decrement Operators

C++ provides ++ and – – operators which are not provided by many other languages. These operators are called increment and decrement operators respectively.

The increment operator ++ adds 1 to its operand.

++ x means $x + 1$

and – – x means $x - 1$

Binary Operators

The Binary operators act upon two operands. The following arithmetical operators require two operands.

Symbols	Meaning	Example
+	Addition or Plus	A + B
–	Subtraction or Minus	A – B
*	Multiplication	A * B
/	Slash or Division	A/B
%	Modulus or Remainder	A%B

Relational Operators

Relational operators are used to determine the relationship between different operands. These are used in the work of comparison also. There are two mathematical expressions and these are connected by a relational operator. The relational expression return 0 if the relation is false and returns 1 if the relation is true. The six relational operators are shown in the following table:

Symbol	Meaning
>	Greater than
<	Less than
> =	Greater than, equal to
< =	Less than, equal to
= =	Equal to
! =	Not equal to

Logical Operators

Logical operators combine the results of two or more than two expressions. The mode of connecting relationship in these expressions refers as logic and the expressions are called logical expressions.

The logical expression returns 1 if the result is true and it returns 0 if the result is false.

C++ provides three logical operators which are listed below.

Symbol	Meaning
!	Logical negation (Logical inverse)
||	Logical OR
&&	Logical AND

STRUCTURE OF A C++ PROGRAM

The following is the structure of a simple C++ Program.

Include Files
Class Declaration
Class Functions Definitions
Main Function Program

```
#include <iostream.h>
main()
    {
    int x,y, sum;                        //Read the values x and y
    cout<<"\Enter the first number:";  //console output and prompt.
    cin>>x;
    cout<<"\Enter the second number:";
    cin>>y;
    sum=x+y;
    cout<<sum<<endl;  //display the output
    return();
    }
```

Explanation

1. A C++ program starts with function called main(). The body of the function is enclosed between curly brackets ({ }). A C++ program can consist of as many functions as required.
2. The explanation written after the characters // is known as comment. Comment plays an important role in development of program. It increases readability of the program.
3. The statement in the program `#include<iostream.h>` is called an **"Include Preprocessor Director."** The symbol # is called hash symbol. The name of the file appears before the start of the program and is known as a **Header File** and has extension *h*.
4. The function **cin** is called console input. It is used with redirection operator >>. It accepts data from the input unit, *i.e.*, keyboard.

   `cin>>x;`

   When this statement is executed, it accepts the value of '*x*' from keyboard.
5. The function cout is called console output. It is used with redirection operator <<. It displays the output on computer screen.

   `cout<<sum<<endl;`

   When this statement is executed, it displays the value (sum) on the screen. endl; is used to show the end of line. It is optional to use endl;

   endl is a manipulator which is frequently used. It has the same effect as character "/*n*", but due to its clarity, endl is preferred.
6. Every statement in C++ should be followed by a semi-colon (;). In case this semi-colon is missed, an error message appears while compiling the program.

Importance of iostream.h File

In every C++ program, it is essential to add `iostream.h` header file.

The following sequence is used in the program.

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

This sequence helps the preprocessor to add the contents of `iostream.h` file to the program. This header file should be included at the start of the program that uses input and output statements. The header file `iostream.h` contains declaration for the identifier cout and for the operator <<.

The cin Operator(>>)

The cin operator is used to get input from the standard input device *i.e.*, keyboard. When you write C++ programs and use cin, you do not know the values of the input until the program is run. You can enter the values from the keyboard after the program is run. You can also fill the values using assignment statement such as $i = 15$; The difference between an assignment statement and cin operator is that the value of input is known in case of assignment statement whereas it is not known in case of cin operator.

Syntax

```
cin>>values;
```

Look at the following program:

```
//INTRO.CPP
#include<iostream.h>
main( )
{
    int x,y;
    cout<<"Enter one value=";
    cin>>x;                        //input the value of x
    cout<<"Enter second value=";
    cin>>y;                        //input the value of y
    cout<<"The sum is=" <<x+y<<endl;
    return0;
}
```

When you will run the above program, computer will ask you to input the values of x and y.

Enter one value = 10

Enter second value = 20

The sum is = 30

The cout Operator(<<)

The << operator is used to obtain the output on a standard output device. This device may be screen of the monitor.

Syntax

```
cout<<data;
```

Examples

```
cout<<"I am a student of B.Com. class";
cout<<"I read in S.D. College";
```

CASCADING OF I/O OPERATORS

The input and output redirection operators are called I/O operators and written as cin >> and cout<<. When a second I/O redirection operator is written after the first I/O redirection operator, it is called cascading of I/O operator.

```
cout <<"The sum of variables"
     <<a<< "and"
     <<b<< "is"<<a+b;
 cin >>"enter the first number">>a;
cout <<"the first number"<<a;
```

The external file `iostream.h` provides input and output functions.

Multiple use of input (>>) and output (<<) operators in one statement is called cascading of I/O operators.

The setw() Manipulator

The `setw()` manipulator is used along with output operator << to display the output. It stands for set width.

This manipulator `setw(n)` puts the output in a specified zone of n characters.

Syntax

setw(int *n*)

The default width is zero.

For example

cout<<set(5) <<50;

The above line segment will set a zone of 5 columns. The number 50 will be displayed as

5 0

Illustration: *The following program illustrates the use of setw ( ) manipulatior.*

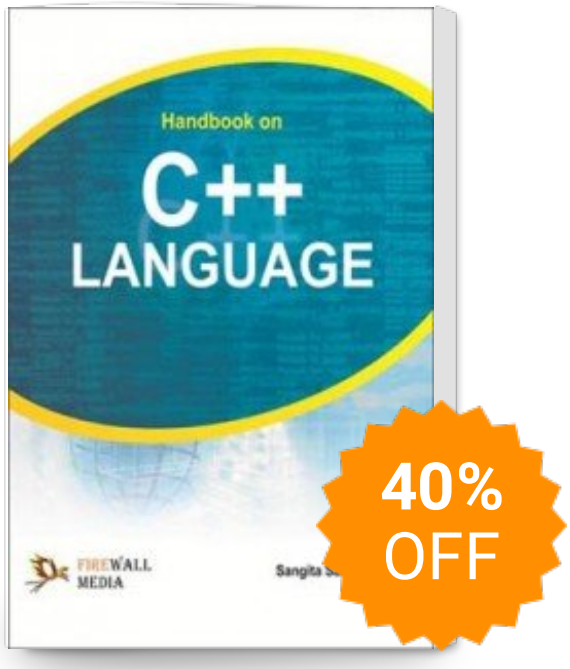
Solution.

```
#include<iostream.h>
# include<iomanip.h>
main()
{
    int x,y;
    x = 200;
     y = 400;
cout<<setw(5)<<x<<setw(5)<<y>>endl;
cout<<setw(6)<<x<<setw(6)<<y>>endl;
}
```

The output of the program will be as under:

```
200 400
200 400
```

Handbook on C++ Language By Sangita Sardana



Publisher : Laxmi Publications ISBN : 9789380298672 Author : Sangita Sardana