

C++ Programming Language

In this chapter, you will learn about :

- Computer Program • Introduction • Using float data type
- Header Files and Program Structure • Understanding your Program
- Mathematical Operators • Relational Operators • Assignments and Expressions
- Output of the Program • Switch Statement • For loop • While Statement
- do-while Statement • Arrays

COMPUTER PROGRAM

A program is a set of instructions that tells a computer how to carry out a certain task.

You already know, computers only understand binary language which is composed of 0's and 1's. This is a low level language and very hard to execute, so higher level languages such as; PASCAL, C, C++ etc. were invented to make the job easier.

A compiler is used to convert a program in C++ to a binary or executable file that the computer can understand.

INTRODUCTION

C++ is an object oriented Programming Language developed by Bjarne Stroustrup in early 1980's. It is an excellent language to start with programming. Once the basic concepts of C++ are clear, learning other computer languages becomes easier.

Like English, C++ also has words, space, punctuation etc.

But the similarity ends here because if you leave out a comma or a semi-colon, or use a lower-case letter. Where an upper-case one should be used, it will change the meaning of the code.

C++ consists of the following attributes:

- Tokens
- Keywords
- Symbols
- Comments
- Variables
- Constants & Literals

Tokens

A WORD OR A SINGLE PIECE OF PUNCTUATION IS CALLED A TOKEN. TOKENS ARE SEPARATED BY WHITE SPACE.

```
INT MAIN C
INT A, B, C;
{
C = A + B;
}
```



Keywords

KEYWORDS ARE RELATIVELY SMALL SET OF WORDS THAT HAVE FIXED MEANINGS. KEYWORDS CANNOT BE USED AS NAMES OF VARIABLES, CLASS, FUNCTION, ETC.

EXAMPLE : VOID , INT, FLOAT, DOUBLE, FIT, LYTE, FOR, IF, WHILE ETC.

Symbols

SYMBOLS CONSIST OF ANY COMBINATION OF LETTERS (UPPERCASE OR LOWERCASE), NUMBERS AND UNDERSCORE CHARACTER '(_)'.

THE FOLLOWING ARE THE VALID SYMBOLS:

ABC

M

NAME

MOHIT-KAPOOR

M5 ETC

Comments

COMMENTS ARE USED TO MAKE YOUR SOURCE CODE EASIER TO READ AND UNDERSTAND. THERE ARE TWO WAYS TO INCLUDE COMMENTS IN YOUR CODE.

Example : `/# This is
Multi-line comment */`

The second way to include comments is to put two slashes (||) before a remark.

Example : `|| This is a single line comment.`

Variables

A variable is a piece of data to be stored in the memory. It must be declared before it can be used.

Declaring a variable tells the compiler :

- Its type
- name
- any other special properties if it has.

The format / syntax / structure of a declaration is-
type name-list

Example :

`Int x ; /* x is an integer */`

`Char c ; /* c is a character */`

Data types / variables

Different types of variables require different amounts of memory.



Here, you will study about :

- int
- float
- double

'Int' is used to declare integer type variable (i.e. whole numbers) and takes up 4 bytes of Memory. e.g. 8, 18, 48 etc.

'float' is used to declare variables of type floating point. This means you can use numbers with decimals in your program e.g. 8.96, 9.00 etc. A float require 4 bytes of Memory.

'double' is used to declare numbers with decimals but it has more memory than int or float. A double uses up 8 bytes of memory. e.g, 3.2483192, 555.89523 etc.

Example : *Program to calculate multiple of two numbers using three variables of integer (int) data types.*

```
# include < iostream.h >
int main ( )
{
int x, y, z; // three variables declared.
x = 30;
y = 40 ;
z = (x * y );
count << x << "*" << y << "=" << z <<end 1;
return 0
}
```

This program will display $30 * 40 = 1200$

USING FLOAT DATA TYPE

```
# include <iostream.h>
int main ( )
{
Float x, y, z;
x = 10;
y = 4;
z = (x/y);
count << x << "/" << y << "=" << z << end 1;
return 0;
}
```

This program will display like $10/4 = 2.50$



CHARACTER TYPE DATA

Suppose : you want to write a program that asks your name.

You can't use a data type such as; int, float or double to store names as name comprises of letters. Thus, a new data type called 'char' is used. A char data type can store a character, i.e., a single character enclosed in single quotation marks, e. g 'A' or 'a' etc.

You can declare and initialise a char variable is;

```
char var 1;
var 1 = 'm';
cout << "variable has the value"<< var 1 << end 1;
```

The result of this would be variable has the value m.

How to store Multiple Characters?

You can do this by using an array of characters.

Example : Declare a character (char) array to store a person's name!

```
char name [10] ;
```

This line will let you store a name up to 10 characters in length. Remember; it accepts the character values from 0 to 9.

```
# include <iostream.h>
Void Main ( )
{
Char name [10];
cout << "Enter a name" << end 1;
cin >> name;
cout << "Hello"<< name << end 1;
}
```

HEADER FILES AND PROGRAM STRUCTURE

In C++ Program you have to include the header files to do the required task. Some important files are listed below along with their usage.

```
# include < iostream. h> // to input and output.
# include < string. h> // to accept sequence of characters.
# include < conio. h> // to clear and return the value on the screen.
# include < iomanip. h> // to set the printing space.
# include < ctype. h> // to execute string / library functions.
# include < math. h> // To execute mathematical functions.
```



Example :

1. # include < iostream.n>
2. Int Main ()
3. {
4. Count << "Hello world!" << end 1;
5. return 0;
6. }

UNDERSTANDING YOUR PROGRAM

Line 1, tells the compiler to include a file called "iostream.h" for input and output operations.

Line 2, 'int Main ()', is the header for the Main function where program is written.

Line 3, the body of the function is enclosed in braces, { _ _ _ }.

Line 4, "cout <<" stands for channel OUT. It is an output statement which displays the message enclosed within double quotes onto the screen.

"end 1" tells the computer to go onto a new line after printing.

Line 5, at the end of the function there is a line "return 0;". This tells the computer that when main () is finished, it returns a value.

MATHEMATICAL OPERATORS

Symbol	Operation	Example	Value
+	Addition	4+5	9
-	Subtraction	43-25	18
*	Multiplication	4*7	28
/	Division	9/2	4.5
%	Modulus (it gives the remainder)	20%6	1.2

RELATIONAL OPERATORS

NORMALLY THESE OPERATOR ARE USED WITH CONDITIONAL STATEMENT.

Symbol	Meaning	Example	Value
<	less than	2<8	TRUE
<=	less than or equal to	48<= 29	FALSE
>	greater than	4<2	FALSE
>=	greater than or equal to	9>=2	TRUE
= =	equal to	26 == 16	FALSE
! =	not equal to	20!=9	TRUE

Shorthand Arithmetic Statements

C++ provides a shorthand notation. The Mathematical operators '+' (addition), '-' (subtraction), '*' multiplication, '/' (division) and "%" (Modulus) can be prefixed to the assignment operator (=), as shown in the following examples:

EXAMPLE

Sum + = 1;

Total- = discount;

Rate * = 2;

Per % = 100

EQUIVALENT TO :

sum = sum 1;

total = total- discount;

rate = rate * 2;

per = per % 100;

Postfix and Prefix operators

The increment operator '++', is also called postfix.

Sum ++;

The operator "++" may also be used as a prefix operator:

++ SUM;

X = 4;

Y = x ++;

result in "x" having the value 5 and "Y" having the value 4, where as.

X = 4;

Y = ++ x;

Both the variables will result value 5. This is because –

- "++x" increments the value of "x" before its value is used.
- "x++" increments the value afterwards.

There is also an operator "--", which decrement variables by 1, and which can also be used in prefix or postfix form.

Boolean Expressions and operators

- The expression such as

" 5 > 9"

"1.5! = 2.9"

"5 > = 9" will evaluate the value to "true" or " false"



Such expression can be combined using the logical operator "& &" ("AND") "||" ("OR") and "!" ("NOT"), as in the following example.

Expression	True or false
(6 < = 6) & & (5<3)	False
(6<=6) (5<3)	True
(5! = 6)	True
(5 < 3) & & (6 < =6) (5! =6)	True

Input

You can use a keyword similar to cout, called "cin" to input a variable.

"cin " stands for channel IN.

Example : This program asks the user to enter a value from the keyboard and displays it

```
# include < iostream.h>
int main ( )
{
int a,b,c;
Cout << "Enter two number" <<end |;
Cin >>a >> b;
C = (a+b);
Cout << "The sum is" << C << end |;
return 0;
}
```

Line 6 : will prompt the user to input the value from the keyboard for the variables a and b.

Line 7 : will calculate, sum of two numbers and store in a variable called "c".

Line 8 : will display the result.

The direction of the arrows (<<) for a cout statement are pointing in the opposite direction which implies data is flowing out.

Conditional Statement :

They are of following types :

- If-else if
- Switch-case
- For loop
- while loop
- do-while loop



If-else if statement

It is also known as Decision Making statement;

Example : Program to read the number from the keyboard and check whether the number is positive, negative or zero.

```
#include < iostream. h>
int main ( )
{
int x;
cout << "Enter an integer" << end 1;
ch >> x;
if (x > 0)
{
cout <<"The number is positive! " << end 1;
}
else if (x < 0)
{
cout << "The number is negative! " <<end 1;
}
else
{
cout << "The number is 0! " << end1;
}
return 0;
}
```

OUTPUT OF THE PROGRAM


When you enter a number, the computer first checks to see if the number is greater than zero then the code in the brackets after the 'if statement' will be executed and the program will finish. The computer does not even look at the other options since a true condition has already been met.

If the number is not greater than zero then the computer will check the next option. i.e $x < 0$. If the number is less than zero then computer will print, "That number is negative!"

Finally, if the number is neither positive nor negative than it is equal to zero, and the computer will execute the final else statement.

Example : If you use only the following 'if statement' in your program'

```
cout << "Enter an integer" << end 1;
cin >> x;
If (x>0)
{
```

```
cout << "The number is positive !" << endl;
}
```

Here, in the above program as you enter any positive number, program will display that number and stops.

You can perform logical tests on a number.

Example:

```
if (x>0 && x<10)
{
cout << "x is between 0 and 10" <<endl;
}
```

This will text if x is greater than 0 AND less than 10.

```
if (x<10 || x >500)
{
cout << "x is less than 100 or it is greater then 500" << endl;
}
```

This will test if x is greater than 500 OR less than 100.

SWITCH STATEMENT

It is also known as user's choice or Menu driven statement. This statement allows you to select the option from the menu.

Example :

```
# include <iostream.h >
# include < conio.h >
void Main ( )
{
clrscr ( ); // clear screen
int ch;
cout << "Enter your choice";
cin >>ch;
Switch (ch);
{
```

```
Case 1 : cout << "Sunday";
```

```
break;
```

```
Case 2 : cout <<"Monday";
```

```
break;
```

```

Case 3   : cout <<"Tuesday";
break;
Case 4   : cout <<"Wednesday";
break;
Case 5   : cout <<"Thursday";
break;
Case 6   : cout <<"Friday";
break;
Case 7   : cout <<"Saturday";
break;
default  : cout <<"Invalid choice-re-enter";
        }
        getch ( ); // to get the value onto the screen.
        }

```

FOR LOOP

To repeat the given instruction again and again, 'for loop' is used.

Example : Program to display out the numbers one to hundred.

will you use one hundred lines like this;

```
cout << "1" << end 1;
```

```
cout << "2" << end 1; ---- etc.
```

This is time consuming.

A much quicker way would be to use a 'for loop',

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

```
#include < conio.h>
```

```
void Main ( )
```

```
{
```

```
clrscr ( ); // to clear the screen
```

```
int i;
```

```
For (i = 1; 1<=100; (++)
```


```
{
```

```
cout << i < end;
```

```
}
```

```
getch ( ) ; // to get the value onto the screen.
```

```
}
```



When the 'for loop' is first encountered, 'i' is set to 1, and the value of i is printed and with the cout statement.

Next time 'i' is incremented by one (i++). Now value of i is equal to 2, and the same is printed out. Again i is incremented and the procedure continues as long as 'i' is less than equal to 100 (i.e the program will print numbers from 1 to 100).

WHILE STATEMENT

IT IS ALSO KNOWN AS ENTRY CONTROL LOOP. IT IS A SUBSTITUTION OF A 'FOR LOOP'

EXAMPLE : PROGRAM TO DISPLAY NUMBERS FROM 1 TO 100 USING 'WHILE LOOP',

```
#INCLUDE < IOSTREAM.H>
#include < CONIO.H>
VOID MAIN ( )
{
CLRSCR ( ) ; // TO CLEAR THE SCREEN
INT I ;
WHILE (I< 100)
{
COUT <<I<< END 1;
I++;
}
GETCH ( ) ; // TO GET THE VALUE ONTO THE SCREEN.
}
```

DO-WHILE STATEMENT

It is used when you do not know how many times (iterations) the program will execute. This loop at least execute once as it starts with a word 'do' and then performs the condition defined in the 'while' statement.

For Example :

```
# include <iostream.h>
#include < conio.h >
void Main ( )
{
clrscr ( );
int x;
do
{
cout << "enter a number" << end 1;
cin >>x;
```

```

}
while (x!=0); // while x is not equal to 0
getch( );
}

```

This program will keep asking the user to enter a number unless 0 is entered. The program exists the 'do-while' loop when 0 is entered.

ARRAYS

Arrays are used to store a collection of variables of same type. For instance, you can have arrays that store numbers, words, etc.

The following could be used to declare arrays;

```
int number [12] // This array will store 12 integer numbers.
```

```
char name [10] // This array will store 9 character Plus 10 names.
```

Each element of the array has an index number. This index number is used to access the value of that element.

Suppose we had an array called 'int arr [10]', than;

The ten numbers are :

```
arr [0]
```

```
arr [1] arr [2] arr [3] arr [4] arr [5] arr [6] arr [7] arr [8] arr [9]
```

You can assign values to each element with other variables.

```
arr [0] =10;
```

```
arr [3] =3;
```

These two lines will set the first element of arr equal to 10 and the fourth equal to 3.

You can initialise an array at the same time as you are defining it as follows;

```
Int arr [5]= { 11, 2, 6, 7, 3 };
```

Alternatively, you could input data from the keyboard.

```
For (i=0;i <5; (i++))
```

```
{
```

```
cout << "enter the element" << i << end 1;
```

```
cin >> arr [i];
```

```
}
```

You may initialise each element individually

```
arr[0] = 11;
```

```
arr[1] = 2;
```

```
arr[2] = 6;
```

```
arr[3] = 7;
```

```
arr[4] = 3;
```

Here is a program that will find the sum of an array :

```
# include < iostream.h>
#include < conio.h>
void Main ( )
{
clrscr ( ) ;
int arr [5];
int i, sum =0, // we must initialise sum to zero.
For (i=0; i< 5; i++)
{
cout <<"enter a number" << endl;
cin>> arr [i];
}
For (i=0; i<5; i++)
{
Sum = sum+arr [i];
}
cout << "The sum of the five numbers is" << sum << endl;
getch ();
}
```

Output

Enter a number (input any five numbers from the keyboard). The program will printout.
The sum of the five numbers is (total of five numbers already input).

Points to Remember

- Program is a set of instructions that tells a computer how to carry out a certain task.
- Computer is used to convert a program in C++ to a binary or executable file.
- C++ consists of many attributes like Tokens, Keywords, Symbols, Comments, Variables, Constants.
- Different types of variables require different amounts of memory.
- "Cin" stands for channel IN.
- There are various types of conditional statements like if-else if, for loop, while loop, do-while loop, switch-case.
- While statement is also known as Entry Control Loop.

EXERCISE



A. Tick (✓) the correct option :

1. _____ is /are attributes of C++.

(a) Tokens

(b) Keywords

(c) both (a) & (b)

2. _____ are used to make source code easier to read & understand.

(a) Comments (b) Symbols (c) Tokens

3. _____ is also known as Entry Control Loop.

(a) While statement (c) Do-while statement (c) For statement

B. Fill in the blanks :

1. _____ is also known as user's choice or menu driven statement.

2. To repeat the given instruction again & again _____ is used.

3. _____ is used when you do not know how many times the programs will execute.

4. A word, a single piece of punctuation is called a _____.

5. A _____ is a piece of data to be stored in memory.

C. Tick (✓) the correct statement and cross (X) the wrong ones :

1. C++ is developed by Bjarne stroustrup in 1980's.

2. Variables are used to make your source code easier to read & understand.

3. Symbols consists of any combination of letters, numbers & that underscore letters.

4. If -- else if statement is known as decision making statement.

5. Do-while statement is known as entry control loop.

D. Define the following :

1. Tokens

2. Comments

3. Symbols

4. Keywords

5. Variables

6. Separators

E. Answer the following questions :

1. What is data type? Give an example.

2. What is the difference between $19/6$ & $19/60$.

3. Give the difference between $++a$, $a++$ & $a+ =1$.

4. What is loop? Explain various types of loops.

5. What is meant by nesting of loops? Give an example.

6. Differentiate between :

(a) Character constant & String constant.

(b) Operator & Operand.

ACTIVITY



• Write the following programs in C++ language in your computer lab.

1. Find the Sum of first to natural numbers

2. To create a Fabonacci series.

3. Find the average of marks of first ten students of your class.