10/5/2020

Programing fundamental

Sir: Abdul Manan Niazai

Rifatullah
KHURSAN UNIVERSITY

Programming Fundamental

1. **Computer: A programmable electronic device that can store, retrieve and process data.**
2. **Computer Programming: Computer programming is a way of giving computers instructions about what they should to do next. These instructions are known as codes, and computer programmers write code to solve a problems and perform Some task.**
3. **Programming => Program=>(Set of instructions).**
4. **Algorithm: A step by step procedure for solving a problem in limited amount of time is called algorithm.**
5. **Programming Language: A set of rules, symbols, and special words used to deal or communicate a computer programs is called PL.**

What are Programming Languages

Programming languages: are those Languages through which we can Developed Different Software's.

Names of Some programming languages

1. **C Language**
2. **C++**
3. **Visual Basic .NET (VB.NET)**
4. **Java**
5. **C# (C SHARP)**
6. **JavaScript**
7. **7. Python**
8. **8. Ruby**
9. **9. Assembly**
10. **and many more….**

Types of Computer Languages

- **There are three main types of Computer Languages:**

1. **Low Level Computer Languages (Assembly language, Machine Code).**
2. **Intermediate Computer Language (C and C++).**
3. **High level Computer language (C, C++, C#, Java, Python, Visual Basic …..).**

What is C++?

- **C++ is a high-level programming language developed by Bjarne Stroustrup at Bell Laboratories.**
- **C++ adds object-oriented features to its predecessor, C.**

C++ is one of the most popular programming language for graphical applications, such as those that's run in Windows and Macintosh Computers.

(C++)

1. **C++ is a High Level Programming Language.**
2. **C++ is an Object Oriented Programming Language.**
3. **C++ is the advance Version of C Language**
4. **It Pronunciation is C++ (See Plus Plus)**

C++ History

1. **BCPL(Basic Combined Programming Language) Language was Developed in 1966**
2. **B language was Developed in 1969**
3. **C language was Developed in 1972**
4. **C++ was Developed in 1979 by Bjarne Stroustroup**

Names of Applications which are Created in C++

A. System Software's: like Windows

B. Application Software's: MS Offices

C. Entertainment Software's: Video Games

What is IDE?

IDE(Integrated Developed Environment)

IDE: Are the area where we can put a Programming Languages Code.

IDEs for C++

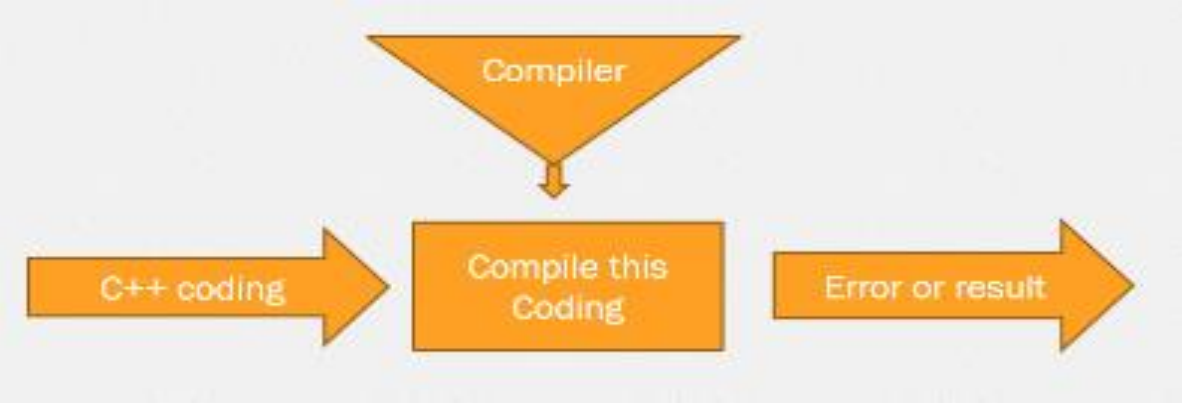
- **DOS BOX**
- **DEV C++.....We will use in this class**

- **Visual C++**
- **Turbo C++ V4.5**
- **And Many more…….!**

What is Compiler?

Complier: A Complier is an important past of a Programming Language which is used to translate / convert Human language into Machine Language.

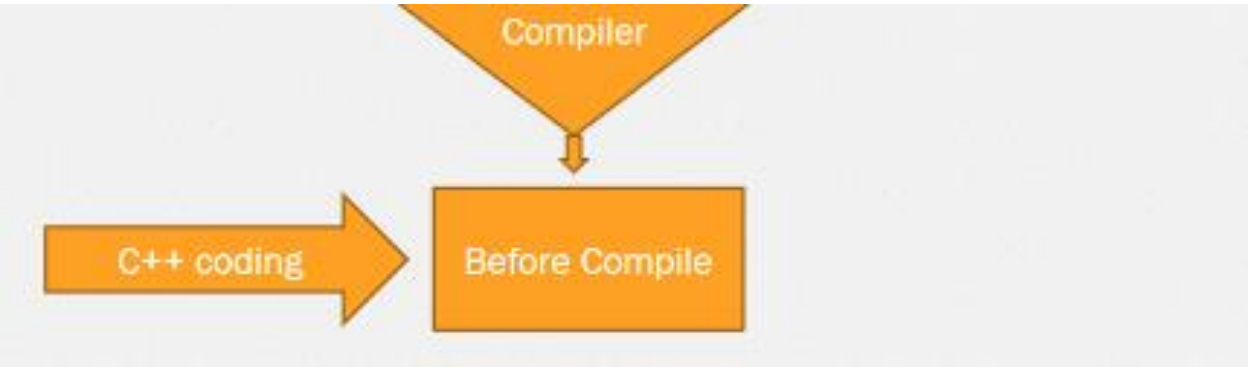
Example:



What is Human like Language?

Human like language is the Raw(Non-processed) C++ statement's.

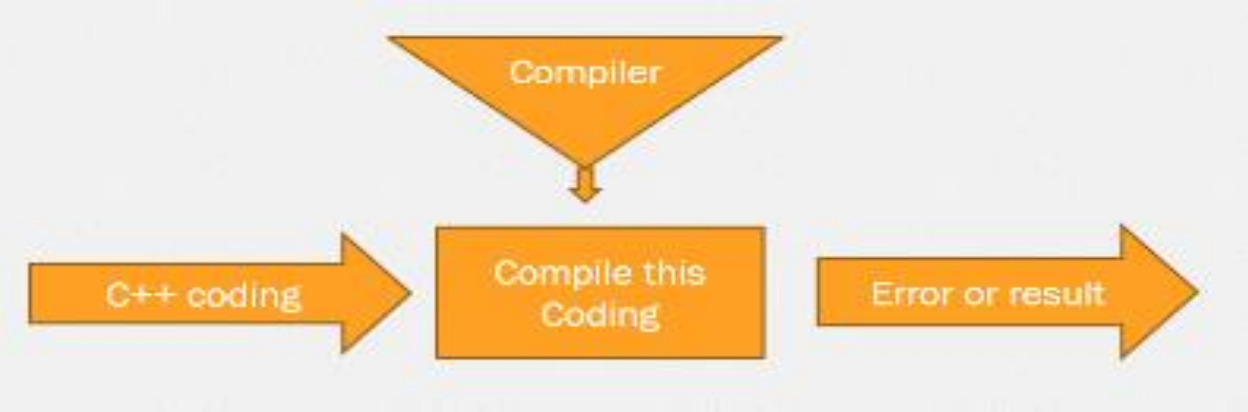
Example:



What is Machine like Language?

Machine like Language is the Meaningful or Processed form of C++ Statements.

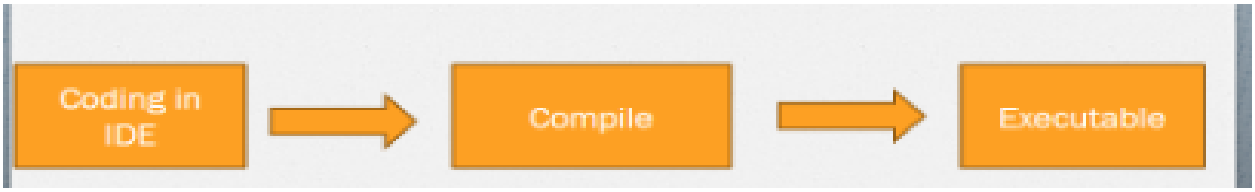
Example:



Rule for C++ Program Execution

C++ follows this steps

- ✓ **Create or Put your Coding in C++ IDE**
- ✓ **Programname.CPP: Save the Program with CPP.Extension**
- ✓ **Compile: After save file then Compile with exe Extension.**



C++ Program Structure

Generally C++ Program have the following Parts

1. **Preprocess Directive(Complier Directive)**
2. **The Main() /// it is Function**
3. **C++ Statements**

Preprocessor Directive

The instructions that are given to the compiler before the beginning of the actual program are called Preprocess Directive. These are also called complier Directive.

Example:

#Include <iostream> // Preprocessor Directive

The Main() Function

The main() function is the important parts of C++ program. If main() function is not included, the program is not complied and it will give Error message.

Syntax of main() function is:

```
main(){
program statements…..
}
```

C++ Statement

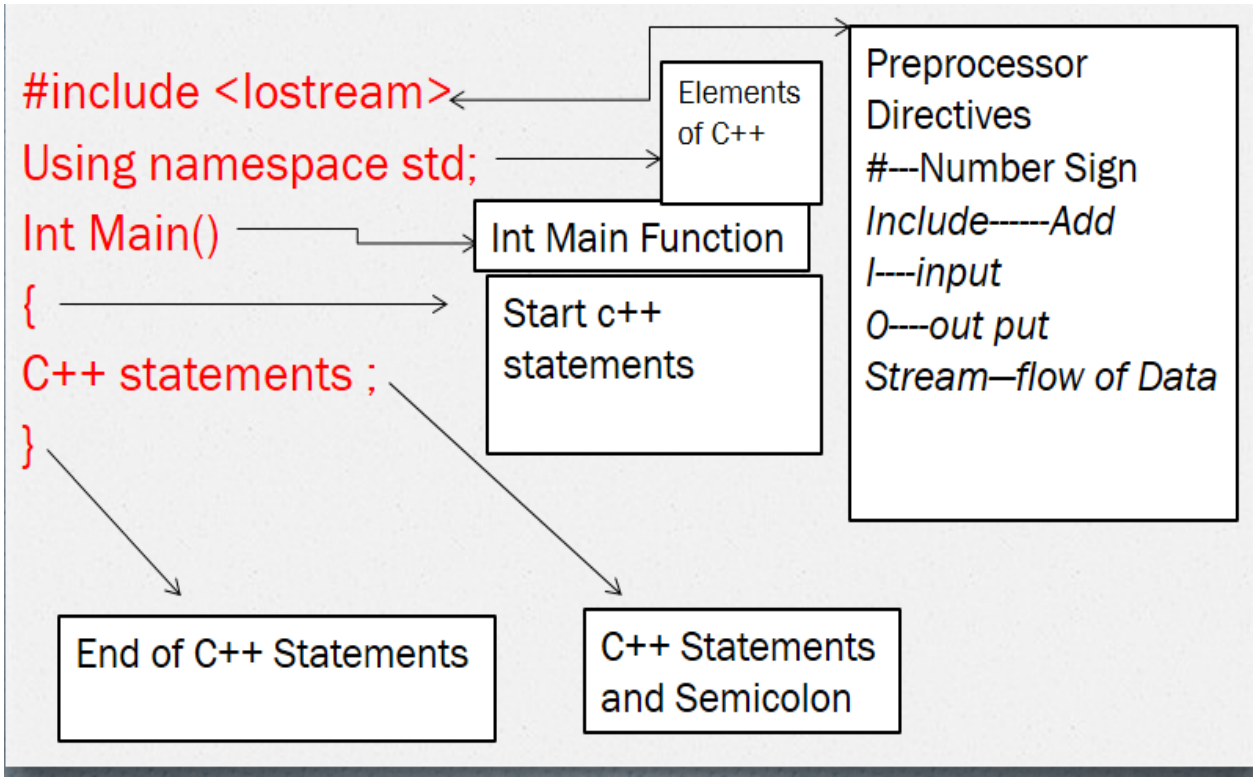
The statement of the program are written under the main() function between the curly braces {}.

These statements are also called program body.

Syntax:

```
{
C++ statement
}
```

Diagram for C++ Structure



What is Keywords?

Keywords : keywords are predefine, reserved words, used in Programming language that have special meaning to the compiler.

These are also called reserved words.

We have 60 keywords in C++ Shown in below..

These are 60 Keywords…

Asm	auto	bool	break	case
catch	char	class	const_cast	continue
default	delete	do	double	else
enum	dynamic_cast	extern	false	float
for	union	unsigned	using	friend
goto	if	inline	int	long
mutable	virtual	namespace	new	operator
private	protected	public	register	void
reinterpret_cast	return	short	signed	sizeof
static	static_cast	volatile	struct	switch
template	this	throw	true	try
typedef	typeid	unsigned	wchar_t	while

What is Variables

A quantity whose value may change during execution of the program is called variable.

A variable represents a storage or memory location in the computer memory.

Data is stored into the memory location.

A variable is also known as object in C++. In C++ A variable name consist of alphabets and digits

Rule for Writing Variable in C++

There are some rules for writing variable in C++:

- **The first character of variable name must be an alphabetic character.**
- **Underscore can be used as first character of variable name.**
- **Blank space are not allowed in a variable name.**
- **Reserved words cannot be used as variable name.**
- **A variable name declared for one data-type cannot be used to declared another data-type.**
- **Data Types in C++**
- **Data types**
- **The variable type specifies the type of data that can be store in it.**
- **OR**
- **Data types define the type of data a variable can hold, for example an integer variable can hold integer data, a character type variable can hold character data etc.**
- **Each variable is declared by it own type.**

  C++ has five basic Data types

  These data types are:

1) **int** **Integer**
2) **float** **Floating Point**
3) **double** **Double Precision**
4) **char** **Characters**
5) **bool** **Boolean**

Int, float, double and char are used in C language. And Boolean is data type a new addition in C++.

Integer Data type

1) **Integer Data type:**

The Int represents the integer data. It is used to declare integer type

of variable.

An integer is a whole number and a number without fraction and decimal point.

For example: 10, 300, 560 and -400, -900 …

Declaration of int Data type: int num = 100;

What is floating point Data type?

A floating point data type variable is a variable that can hold a real number, such as 4320.0, -3.33, or 0.01226.

There are three different floating point data types :

I. **float**

II. **double**

III. **long double.**

The float Data type:

2) float Data type:

The float represent real or floating type of data.

The real type of data is represented in decimal or exponential notation. Float type of data may be signed or unsigned.

Example: 25.21, 34.45, -8.45, 3.59 …

float: For single precision floating point. Size 4 bytes

Declaration of float Data type: float a = 4.45;

Double Data type:

3) Double data type:

The double data type is a real or floating data type. Its twice the capacity of float data type. It is used to store large amount of values.

Double data type: size of 8 bytes

Declaration of double dValue2 = 1.5;

Long double data type:

The long double data type variable are used to store very large real data values. It is also holds decimal values. Storage capacity for long double variable is the 16 bytes.

A long double variable can store real values from 3.4×10^{-4932}

The character Data Type

4) Char data type:

The abbreviation char is used as a reserved keyword in some programming languages, such as C, C++, C#, and Java. It is short for character, which is a data type that holds one character (letter, number, etc.) of data. For example, the value of a char variable could be any one-character value, such as 'A', '4', or '#'.

Declaration of char: char a=' B';

Boolean Data Type:

5) Bool data type:

The Boolean data type is used to declare a variable whose value will be set as true (1) or false (0). To declare such a value, you use the bool keyword.

Example of Boolean:

Example:

```
#include<iostream>

using namespace std;

 main(){

bool input;

cin >> input;

if (input == true){

    cout <<"true you sad";

 }else{

cout <<"false your are Happy!";

} }
```

}

What are Operators

Operator:

An operator is a symbol that tell to the computer to perform certain Mathematical or logical operations. Operators are used in programs to manipulate data and variables.

Types of operators:

- **Arithmetic operators**
- **Assignment operators**
- **Relational operators**
- **Logical operators**

Arithmetic operators:

Arithmetic operators are the symbols that represent arithmetic math operations. Examples: + (addition operator), - (subtraction operator), * (multiplication operator), / (division operator) and % (Modulus or Remainder).

- **An arithmetic operator is a mathematical function that takes two operands and performs a calculation on them**

- **The following arithmetic operators are used in C++:**

Operators	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder

Relational Operators

In C++ Relational operators specify the relation between two variables by comparing them it is also called compression operators.

There are 5 relational operators in C++!

I. >

II. >=

III. ==

IV. !=

V. < or <=

Relational operators:

Relational Operators	
Operator	Meaning
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to

Logical operators:

Logical Operators: Logical Operators are used to combine two or more conditions. These operators are used to perform logical operations on the given expressions. There are 3 logical operators in C++ language. They are, logical AND (&&), logical OR (||) and logical NOT (!). The result of the operation of a logical operator is a boolean value either true or false.

Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	(5<2)&&(5>3)	False
||	Logical or	(5<2)||(5>3)	True
!	Logical not	!(5<2)	True

Basic Input and Output

Input and Output:

The statement that are used to get data and then assign to variable as known as input statements.

The statements that are used to receive data from the computer memory and then send to the output device are called output statements.

The cout object:

The "cout" is pronounced as 'See Cout' the cout stands for console output. The console represents the computer display screen.

The 'cout' is a C++ predefine object. It is used as output statement to display output on the computer screen. It is a part of iostream header file.

Flow of data from one location to another location is called stream.

The 'cout' is the standard output stream in C++. This stream sends the output to the computer screen.

Endl Manipulator:

Endl is a manipulator in C++ which is used to insert a new line to the program. Instead of endl we can used "\n" to insert new line.

Question:

Write a program to print a message on screen by sing "endl" manipulator ?

```
cout<<"I am "<<endl<<"Afghan";

cout<<"I am \n"<<"Afghan";
```

The cin object:

The "cin" object stands for console input. It is pronounced as "See In" it is input stream. It is used as input statement to get input from the keyboard during the execution of the program.

When an input statement is executed, the computer waits to receive an input from keyboard. When a value is typed and enter key pressed the value is assigned to the variable.

The "cin" object is also a part of iostream header file.

Syntax:

```
cin >> vari1;
```

Write a program to get two numbers from keyboard during program execution. Calculate the sum and product of the numbers and then print the result on the screen.

```
Int n1, n2,  sum, product;

Cout<<"Enter first number :";

Cin>>n1;

Cout<<"Enter Second number :";

Cin>>n2;

Sum=n1+n2;

Product=n1*n2;
```

Compound Assignment statement

The assignment statement can also be used to assign one value to many variables. This type of assignment statement is called compound statement.

For example: to assign an integer value 16 to two integer type variables x and y. the compound assignment statement is write:

```
x = y = 16;
```

Write a program to assign a value 40 to integer type variables x, y, a, b, s and c. also calculate the sum of the variables and print the result on the screen.

```
Main(){

Int x, y, a, b, s, c;

x=y=a=b=s=40;
```

c=x + y + a + b + s;

Cout<<"Sum of ="<<c;

}

Compound Assignment Expression

Compound assignment expression is used to add, subtract, multiply or divide a value from variable without writhing the variable on either side of the assignment operator.

In this assignment expression, the arithmetic operator is combined with the assignment operator.

Syntax is:

var op = expression;

For example:

We add 10 to a variable xy that already has a value, the simple arithmetic statement is written as:

xy + = 10;

xy = xy + 10 ;

Following are the some examples of compound assignment expressions:

X += 8 ; // same as: x = x + 8;

X -= 3 ; // same as: x = x -3;

X *= 6 ; // same as: x = x * 6;

X /= 4 ; // same as: x = x / 4;

Write a program in C++ to assign three values to three integer type variable a, b & c. add variable a & b and multiply their sum to variable c. use compound assignment statement.

Main(){

Int a, b, c;

a = 3 ;

b = 6 ;

c = 9 ;

c * = a + b;

Cout<<" Result = " <<c ;

}

- **Increment & decrement operators The operator that is used to add 1 to the value of a variable is called increment operator.**
- **Similarly, the operator that is used to subtract 1 from the value of a variable is called decrement operator.**

The Increment Operator (++)

The increment operator is represented by a double (++) sign. It is used to add 1 to the value of an integer variable. This operator can be used before or after the variable name.

For example, to add 1 to a value of variable xy, it is normally written as: xy = xy + 1 ;

By using increment operator "++" it is written as: xy++;

The increment operator can be written either before or after the variable. If it is written before the variable it will called prefixing. If it is written after the variable it will called post fixing. Prefix and postfix operators have different effects when they are used in expressions.

The Decrement Operator (--)

The decrement operator is represented by a double minus (--) sign.

It is used subtract 1 from the value of an integer variable.

And decrement operator also used prefix and postfix are same like in increment operator.

For example, to subtract 1 from the value of a variable xy the decrement statement is written as:

xy -- ; or -- xy ;

The comment statement

The comment statement is a non-executable statement. It is used to remarks or comments in a program. The comments are usually given to the explain the logic of the program. The complier ignore the comments given in the program.

In C++ double slash's " // " used to mark the comment statement.

For example:

```
// this is our first program

  // java like c++ program language
```

Escape Sequences

What is Escape Sequences

Escape sequences are special characters the backslash (\) symbol considered on escape character because it causes on escape from normal interpretation of a string so that the next character has special meaning. For example, you want to put a new line and break the current line in output of C++ statement then you will use "\n" character which is an escape sequence itself.

An escape sequence consists of two or more characters. For all sequences, the first character will be backslash ("\"). The other characters determine the interpretation of escape sequence. For example, "n" of "\n" tells cursor to move on the next line.

These are some example:

New Line (\n):

Backspace (\b):

Tab (\t):
Alert Bell (\a):

Single Quote (\'):

Double Quote (\");

Double backslash (\\)

Conditional Statements

Conditional statements: The statements of a computer program are executed one after the other in the order in which they are written. This is known as sequential execution of a program. But the order of execution of the statements in a program can be changed. This is done with the help of condition statements.

The condition statements are used to execute or ignore a set of statements after testing a condition.

Conditional statements, also known as selection statements, are used to make decisions based on a given condition. If the condition evaluates to True, a set of statements is executed, otherwise another set of statements is executed.

Conditional statements

- **If statement**
- **If else statement**
- **If else if statement**
- **Nested if statement**
- **Switch statement**

- **If Statement**

- **The " if statement" is used to execute (or ignore) a set of statements after testing a condition.**
- **The " if statement" evaluated a condition. If the given condition is true. The statement following the "if statement" is executed. If the given condition is false, the statement following the "if statement " condition is ignored and the control transfers to the next statement.**
- **The syntax of the "if statement" is:**
- **If (condition) {**
- **Statement – 1**
- **Statement – 2**
- **Statement – m**
- **}**
- **Statement-n**
- **The above syntax. The set of statement (from 1 to statement – m) has been enclosed in curly braces. These are called compound statements.**
- **If the condition given in the "if statement" is true. The compound statement given in the curly braces are executed. If the condition is false. The statement shift to the statement – n and the set of statements that follows the "if statement" is ignored.**
- **The below example execute a single statement if the given condition is true.**
- **#include<iostream>**
- **Using namespace std;**
- **Main(){**
- **Int a,b;**
- **a=10;**
- **b=5;**
- **If(a>b){**
- **Cout<<"We love Afghanistan";**
- **}**
- **}**
- **The following program executes a set of statement when the given condition is true.**
- **#include<iostream>**

- **Using namespace std;**
- **Main(){**
- **int a;**
- **cin>>a;**
- **if(a<=20){**
- **Cout<<"Kabul"<<endl;**
- **Cout<<"Nangrahar"<<endl;**
- **Cout<<"Logar"<<endl;**
- **}**
- **write a program to input two integer values and find the out the first or second number is greater.**
- **main(){**
- **int x,y;**
- **cout<<"Enter your first No:";**
- **cin>>x;**
- **cout<<"Enter your second No:";**
- **cin>>y;**
- **if(x>y)**
- **cout<<"First value is greater";**
- **if(y>x)**
- **cout<<"Second value is greater";**
- **}**

If else Statement

This is another form of the if statement. It is used for making two way decision. In this statement one condition and two blocks statements are used.

The if-else statement tests the given condition. If the condition is true the first block of statement is executed. If the given condition is false the first block of statement is ignored and then will execute of else statement block.

The syntax of if-else statement is:

Syntax:

```
if (Condition)
{
        statement -1
        statement -2
        statement –m
}
else
{
        statement -1
        statement -2
        statement -n
        }
```

Write a program to input a number from the keyboard use if-else statement to find out whether the number is less than or greater than 50.

```
Main(){
Int x;
Cout<<“Enter an integer value : “;
cin>>x;
If(x>50){
cout<<“Number is greater than 50 “;
}
else
{
cout<<“Number is less than from 50”;
}
```

Write a program to input two integers values and then find out which these numbers are equal or different.

```
main(){

int a,b;

cout<<"Enter first Number: ;

cin>>a;

cout<<"Enter second Number: ;

cin>>b;

if(a%2 == b%2){

cout<<"both values are equal ";

} else{

cout<<"values are different ";

}
```

Nested if Statement

When an " if statement" is used within another " if statement " it is called " nested if statement ". The nested if statement is used for multi way decision making.

Syntax of nested if statement is:

```
if(condition -1)
{
        if(condition -2)
        {
                statement -1
        }
        else{
                statement -3
                }
}
```

statement-3

Write a program to input three integer values. Compare the three values to find out if they are equal use "nested if statement" and print the message "All values are equal " . Otherwise print the message "these are different".

```
#include<iostream>

Using namespace std;

Main(){

Int a,b,c;

Cout<<"Enter first value ?";

Cin>>a;

Cout<<"Enter second value ? ";

Cin>>b;

Cout<<"Enter third value ? ";

Cin>>c;

If( a==b){

        if(a==c){

        cout<<"All values are equal";

        } else{

        cout<<"Values are not equal";

}
```

write a program to input three integer values from the keyboard find out the greatest number and print it on the screen.

```
#include<iostream>

Using namespace std;

Main(){

Int a,b,c;

Cout<<"Enter first value ?";

Cin>>a;
```

```
Cout<<"Enter second value ? ";
Cin>>b;
Cout<<"Enter third value ? ";
Cin>>c;
        if(a>b)
        {
                if(a>c)
                {
                cout<<"1st number is greater";
                }
                else{
                cout<<"2nd number is greater";
                }}
        else {
                if(b>c){
                cout<<"2nd number is greater";
                }
                else{
                cout<<"3rd number is greater";
                }
```

If else if statement

if-else-if Statement

if-else-if statement is used when we need to check multiple conditions. In this control structure we have only one "if statement" and one "else statement", however we can have multiple "else if statement" blocks. It is also called if ladder statement.

Its General syntax:

```
if(condition_1)
```

{

Statement -1

}

else if(condition_2)

{

Statement -2

}

else if(condition_3)

{

Statement 3

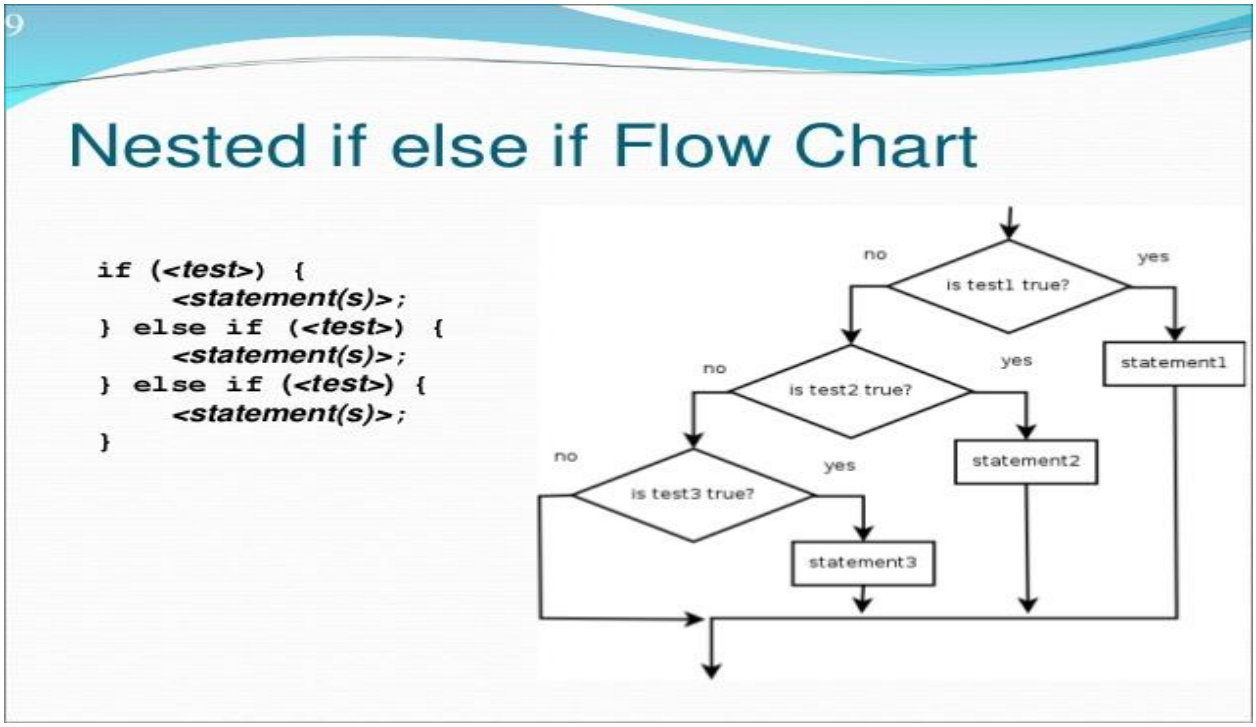
}

else {

if above all conditions are false then will execute this statement

}

If else if flow chart



Write a program to enter two (x and y) numbers to check which one of these greater or equal.

```
#include<iostream>
using namespace std;
main(){
        int x,y;
        cout<<"Enter Number for X: ";
        cin>>x;
        cout<<"Enter Number for Y: ";
        cin>>y;
if(x>y)
{
cout<<" X is greater than Y ";
}
else if(y>x)
{
cout<<" Y is greater than X ";
}
else
{
cout<<" Both X and Y are Equal";
}
}
```

Write a program that input test score of a student and display his grade to the following criteria.

test score	Grade
>=90	A
80-89	B
70-79	C

60-69 D

Below-60 F

```
#include<iostream>
using namespace std;
main(){
	int score;
	cout<<"Enter Score: \n";
	cin>>score;
	if(score>=90){
		cout<<"'A'";
	}

else if(score>=80){
		cout<<"'B'";
	}
	else if(score>=70){
		cout<<"'C'";
	}
	else if(score>=60){
		cout<<"'D'";
	}
	else{
		cout<<"'Fail'";
	}
}
```

Switch statement

The switch statement is used as substitute of nested if else statements. It is used when multiple choices are given and one choice is to be selected.

The nested if else structure become complicated in multiple choices. The switch statement is used in such situations. Only one condition is given in the switch statement and multiple choices are given inside the main body as cases.

Syntax of switch is:

```
switch(expression or variable)
        {
             case val-1: // colon not semicolon
             statement-1;
             break;

             case val-2: // colon not semicolon
             statement-2;
             break;
             - - - - - -
             - - - - - -
             case val-n:
             statement-n;
             break;
             default:    // colon not semicolon
             statement;
        }
```

Break statement

The break statement is used to exit from the body of the switch structure.

In the switch statement the break statement is normally used at the end of statement in each case. It exit the control from the body of switch structure.

Different between nested if else and Switch

	Nested if else	Switch Case
1)	It becomes complicated for multiple selection.	1) it is easy to understand for multiple selections.
2)	If else statement can be used any types of expression like integer, float	2) switch case can test only integer and character expression.

character.

3) Else can be used outside of if body. 3) break can be used inside of switch body.

4) Either if statement will be execute 4) switch statement can execute one by one

or else statement is execute. case after using break.

Write a program to input an integer value. Test the integer value if the value is divisible by 2. then print the message " Divisible by 2 " otherwise "Not divisible by 2 " by using switch statement.

```
#include<iostream>

using namespace std;

main(){

int x;

cout<<"Enter any value";

cin>>x;

switch(x%2)

{

        case 0:

        cout<<"Divisible by 2 ";

        break;

        case 1:

        cout<<"Not divisible by 2";

        break;

}

        cout<<"OK";

}
```

Question 2 in switch:

Write a program to input a character value to check the case if the case is true then will execute the same case.

```
#include<iostream>

using namespace std;

main(){

char x;

cout<<"Enter value";

cin>>x;


switch(x){

        case 'a':

        cout<<"Excellent";

        break;

        case 'b':

        cout<<"Very Good";

        break;

        case 'c':

        cout<<"Good";

        break;

                default:

                cout<<"Invalid input";

                break;

        }


}
```

Goto Statement

The goto statement is an unconditional control transfer statement. It is used to transfer the control to a specified label in the same program without evaluating any condition.

Syntax :

goto label;

label:

Label:

Represent the label in the program to which the control is to be transferred.

Label can be any name followed by a colon. It is given anywhere in the program. And any word that can be used as a variable name can be used as a label in a c++ program

Write a program to transfer the control from one statement to another statement by using " goto statement".

```
#include<iostream>
using namespace std;
main()
{
    cout<<"1st program";
    cout<<"2nd program";
```

- **goto label;**
- **cout<<"\n3rd program";**

```
cout<<"\n4th program";
cout<<"\ n5th program";
Label:
cout<<"\n6th program";
cout<<"\n7th program";
cout<<"\n8th program";
}
```

Write a program to print first ten natural number. Using "goto and "if statement".

```
#include<iostream>
using namespace std;
main(){
int a=1;
```

```
abc:
cout<<a<<endl;
a++;
if(a<=10){
goto abc;
}
```

Loops Statements

A statement or a set of statements that is executed repeatedly is called loop. The statements in a loop are executed for a specified number of time or until some given condition remains true.

Basically loops are used for two purposes:

- **To execute a statement or number of statements for specified number of time. For example a user may display his name on screen for 10 times.**
- **To use a sequence of values. For example a user may display a set of natural numbers from 1 to 10.**

Types of loop:

In C++ there are three kinds of loop statements. These are:

- **while loop**
- **do while loop**
- **for loop**

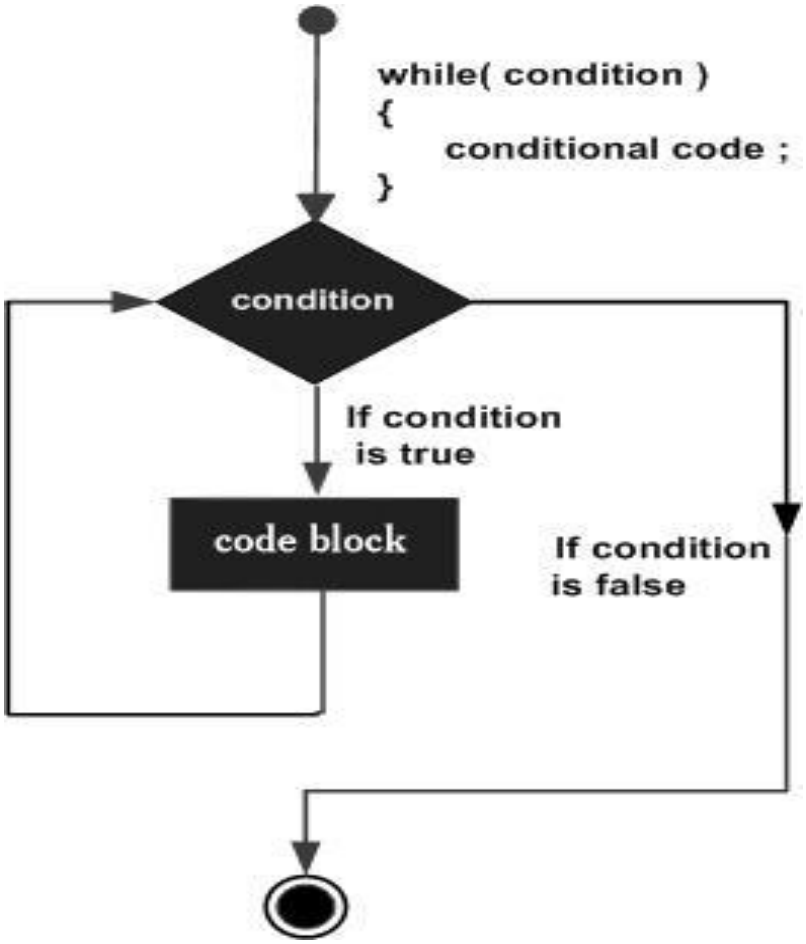
"while" Loop

It is a conditional loop statement. It is used to execute a statement or a set of statements as long the given condition remains true.

The syntax of the while loop is:

```
while (condition)
{
        statement;
}
```

Flow chart of while loop:



Write a program to print five time our country name using while loop.

```
#include<iostream>
Using namespace std;
Main()
{
        int c;
        c=1;

while (c<=5)
cout<<" Afghanistan "<<endl;
c=c+1;
}
cout<<" Program Ends ";
}
```

Question 2 in while loop:

Write a program to print the multiplication table of a number entered from the keyboard.

```
#include<iostream>
Using namespace std;
Main()
{
        int tab, res, a;
Cout<<"Enter the value of table";
Cin>>tab;
a=1;
While(a<=10){
        res=tab*a;
        cout<<tab<<"*"<<a<<"="<<res<<endl;
        a++;
}
}
```

Do while loop

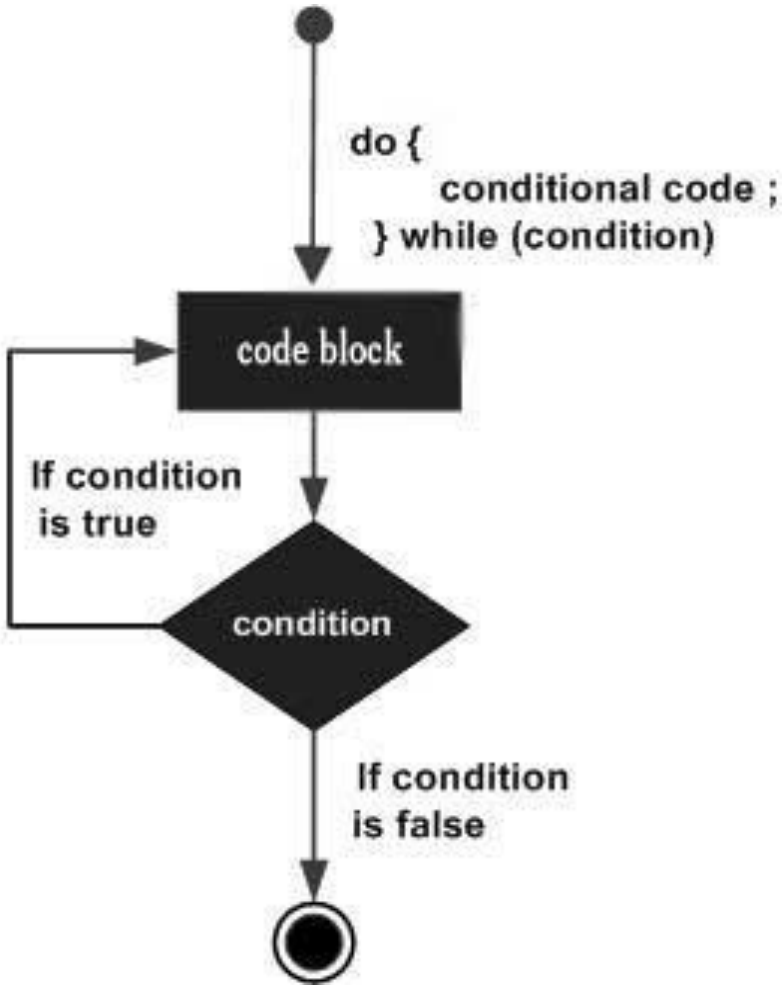
The do while loop is also a conditional loop statement. It is like while loop but in this loop the condition is test after executing the statement of the loop.

In do while loop the body of the loop is executed at least once before the condition is test. It is repeatedly executed as long as the test condition is true.

The syntax of do while loop is:

```
do
{
        statements;
}
while (condition);
```

Flow chart of do while loop is:



Write a program to print ten natural number on the computer screen by using "do-while" loop.

```
main(){
        int n;
        n=1;
        do
        {
                cout<<n<<endl;
                n++;
        }
        while(n<=10);
        }
```

Write a program to input and print the record of a student. Repeat this process for other students until the given condition remains true.

```
#include<iostream>
using namespace std;
main(){
        int marks;
        string name, address;
        char op;

do{
        cout<<"Enter Student name: ";
        cin>>name;
        cout<<"Enter address Marks: ";
        cin>>address;
        cout<<"Enter Marks: ";
        cin>>marks;
        cout<<endl<<"std record is:"<<endl;
        cout<<"name"<<"\t"<<"address"<<"\t"<<"marks"<<endl;
        cout<<endl<<name<<"\t"<<address<<"\t"<<marks<<endl;
        cout<<endl<<"more records [y|n]";
        cin>>op;
}while(op == 'y' || op=='Y');
cout<<"wrong input";
}
```

For loop

The for loop statement is used to execute a set of statements repeatedly for a fixed number of time. It is also known as counter loop.

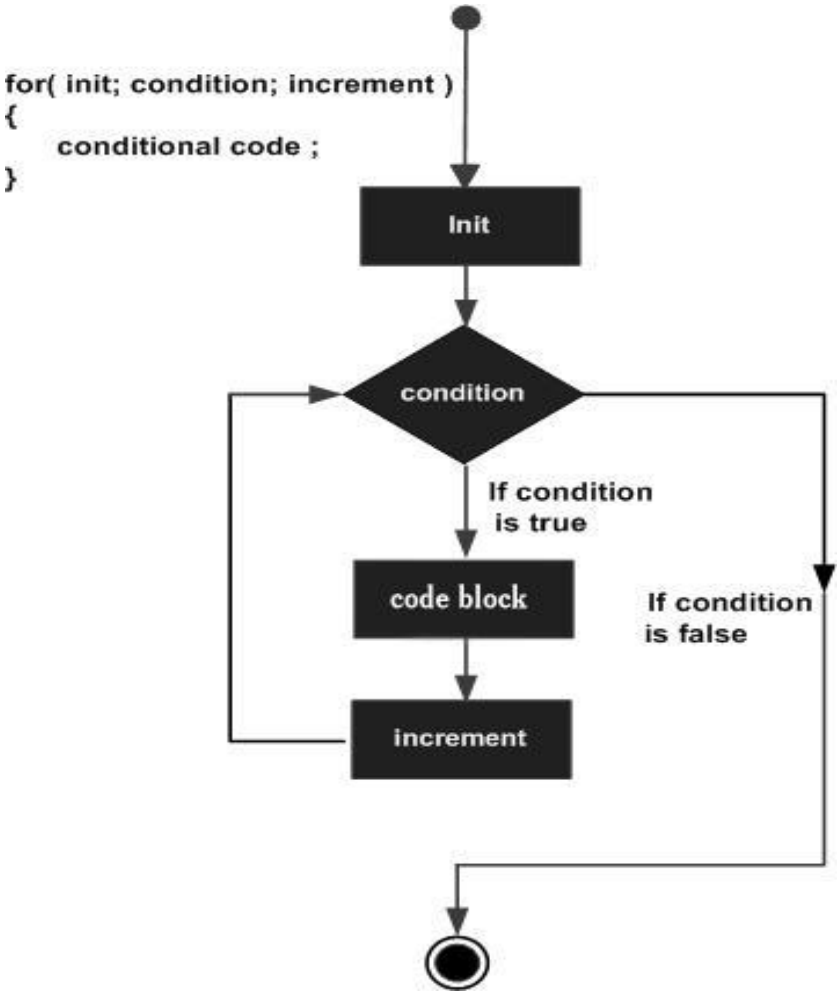
The structure of this loop is different from both the while and the do while loops.

The general syntax of the for loop is:

```
for(initialization; condition; update)
        {
```

}

Flow char of for loop is:



Write a program to print five times natural number using for loop.

```
main(){
    int c;
    for(c=1; c<=10; c++)
    {
        cout<<c<<endl;

    }
```

Write a program to print a table of a given number using for loop.

```
main(){
    int tab, x,r;
```

```
cout<<"Enter any values";
cin>>tab;
for(x=1; x<=10; x++)
 {
        r=tab*x;
        cout<<tab<<"x"<<x<<r<<endl;
}
```

Write a program to calculate the sum of odd numbers from 1 to 10 and then print the sum on the screen.

```
main(){
        int n,sum;
        sum=0;
        for(n=1; n<=10; n=n+2) {
        sum = sum + n;
        cout<<n<<endl;
        }
        cout<<"sum of ="<<sum<<endl;
        }
```

Write a program to print " Khurasan University " ten times using " while", " do while " and " for while " loops.

Jumping in loops

Jumping of two type

1. **Continue**
2. **Break**

The continue statement shift the control back to the beginning of the loop. It is used inside the body of the loop. When this statement is executed inside the body of the loop, the control shift to the first statement of the body of the loop.

Continue statement

Example:

```
#include<iostream>
using namespace std;
```

```
main(){
for(int a=1;a<=10;a++){
cout<<“Afghanistan"<<endl;
continue;
cout<<“Pakistan"<<endl;
}
}
```

Break Statement

The break statement terminates the execution of the loop when it is used inside the body of the loop.

Different between break statement and continue statement

- **The break statement terminates the loop during the execution.**
- **The continue statement is used to shift the control at the beginning the loop.**
- **Example of:**

Both are continue and break statements.

```
#include<iostream>
using namespace std;
main(){
for(int a=1;a<=10;a++){
    if(a==4){
        continue;
    }
    if(a==7){
        break;
    }
cout<<a<<endl;
}
}
```

Nested Loops

The Nested Loops

A loop structure completely inside the body of another loop structure is called nested loop.

Nested loops can be while, do while and for loop.

Example:

Syntax of nested while loop:

```
while(outer-condition)
{
        outer statement –n;
        while(inner- condition)
{
         inner statement –n;
}
        outer statement –n;
}
```

```
#include<iostream>
using namespace std;

main(){
    int u , i ;
    u=1;
    while(u<=2)
    {
        I =1;
        cout<<u<<endl;
        while( I <=3){
            cout<<“Afghanistan"<<endl;
            I ++;
    }
    u++;
    }
    }
```

Output is:

1

Afghanistan

Afghanistan

Afghanistan

2

Afghanistan

Afghanistan

Afghanistan

Example:2

```
#include<iostream>
using namespace std;
main(){
	int u,i,n;
	u=1;
	while(u<=3)
	{
		n=1;
		i=1;
		while(i<=5)
	{
		cout<<n<<"\t";
			n=n+u;
			i=i+1;
	}
```

```
        cout<<endl;
        u=u+1;
}
}
```

Output is

```
1   2   3   4   5
1   3   5   7   9
1   4   7   10  13
```

Nested For Loops

```
main(){
int x,y;
for(x=1; x<=5; x++)
{
        for(y=1; y<=x; y++)
        {
                cout<<y<<"\t";
        }
        cout<<endl;
}
```

Output is

```
1
1    2
1    2    3
1    2    3    4
1    2    3    4    5
```

Example 2 of Nested for loop:

```
#include<iostream>
using namespace std;
main(){
        int x,y;
        for(x=1; x<=5; x++)
        {
                for(y=1; y<=x; y++)
                {
                cout<<"*";
                }
                cout<<endl;
        }
```

Thanks.

Writing by Rifatullah (Ahsan)