م / مراجعة مهمه اساسيات البرمجة

1. Variables in C++

Variables are used to store values. Variable name is the name of memory location where value is stored. It must be alphanumeric, only underscore is allowed in a variable name. It is composed of letters, digits and only underscore. It must begin with alphabet or underscore. It cannot be begin with numeric.

Declaration of Variable

Declaration will allocate memory for specified variable with garbage value.

Syntax :

```
Data-Type       Variable-name;
```

Examples :

```
int a;
float b;
char c;
```

Initialization of Variable

Initialization means assigning value to declared variable. Every value will overwrite the previous value.

Examples :

```
            int  a = 10;
            float  b = 4.5;
             char  c = 'a';
              Character value must be enclosed with single quotes.
a=4.5;
```

if we assign decimal value to integer variable, it will accept only integer portion of value. In the above example variable a will accept 4 only.

Not:

- The variable list may consist of one or more identifier names separated by commas. Some **valid declarations are shown here**:

Example:

```
int    i, j, k;
char   c, ch;
float  f, salary;
double d;
```

The line **int i, j, k;** both declares and defines the variables i, j and k; which instructs the compiler to create variables named i, j and k of type int.

- Variables can be initialized (assigned an initial value) in their declaration. The initializer consists of an **equal sign** followed by a **constant expression** as follows:

Example:

```
int    d = 3, f = 5;  // definition and initializing d
and f.
float  A1 = 2.2;     // definition and initializes 2.2
char    x = 'x';      // the variable x has the value
'x'.
```

1.1 C++ Program Structure

Let us look at a simple code that would print the words Hello World.

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

int main()   // main() is where program execution begins.
{
cout << "Hello World";    // prints Hello World
return 0;
}
```

Let us look at the various parts of the above program:

1. The C++ language defines several headers, which contain information that is either necessary or useful to your program. For this program, the header **<iostream.h>** is needed for output string in the screen.
2. **int main()** : is the main function where program execution begins.
3. // : is a single-line comment available in C++. Single-line comments begin with // and stop at the end of the line.
4. **cout << " : This is my first C++ program.";** causes the message "This is my first C++ program" to be displayed on the screen.
5. << : it is the send operator
6. **return 0**: terminates main() function and causes it to return the value 0 to the calling process.
7. **;** : semicolon , its used as terminator for every C++ statement.

❖ The **OUTOUT** for this program is :

```
Hello World
```

1.2 Standard Output (cout)

cout: the standard output of a program is the screen, and the C++ stream object defined to access it is cout. The < < **operator** is overloaded to output data items of built-in types integer, float, double, strings and pointer values.

Example:

```
cout << "Output sentence";    // prints Output sentence on
screen
cout << 120;                  // prints number 120 on screen
cout << x;                    // prints the content of x on
screen
```

1.3 Standard input (cin)

cin: is the input stream object, its read the input value from keyboard.
>> : it is the operator use to get from operator.
endl : is used to add a new-line at the end of the line.

- You can also use cin to request more than one datum input from the user:

cin >> a >> b;

is equivalent to:

cin >> a;
cin >> b;

- In both cases the user must give two data, one for variable a and another one for variable b that may be separated by any valid blank separator: a space, a tab character or a newline.

Example:

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

int main()
{
char name;
cout << "Please enter your name: ";
cin >> name;
cout << "Your name is: " << name << endl;
return 0;
}
```

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

int main()
{

 cout << "This is a sentence,";

cout << "This is another sentence.";

return 0;
```

❖ The **OUTPUT** for this program: will be shown on the screen one following the other **without any line break between them**:

```
This is a sentence,This is another sentence.
```

Example:

```
#include <iostream>
using namespace std;
int main ()
{
int i;
cout << "Please enter an integer value: ";
cin >> i;
cout << "The value you entered is  " << i;
cout << " and its double is " << i*2 << endl;
return 0;
}
```

❖ The **OUTPUT** for this program: will be shown on the screen:

```
Please enter an integer value: 702
The value you entered is  702 and its double is 1404.
```

C++ Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical calculations on operands(variables).

Types of operators available in C++

- Arithmetic / Mathematical operator
- Assignment operator
- Increment Decrement operator
- Relational operator
- Logical operator
- Unary operator

Arithmetic Operator:

There are following arithmetic operators supported by C++ language:
Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiplies both operands	A * B will give 200
/	Divides numerator by de-numerator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0

Increment Decrement operator

Increment Decrement operators increase or decrease the operand by one value.

Example: Assume A=10, find the output result for the following expressiones:

++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

Assignment operator

Assignment operator is used to copy value from right to left variable.

```
Suppose we have:
float X = 5, Y = 2;
```

=	Equal sign Copy value from right to left.	X = Y, Now both X and Y have 2
+=	**Plus Equal operator** to increase the left operand by right operand.	X+=5 → X=X+5 will give X= 10
-=	**Minus Equal operator** will return the subtraction of right operand from left operand.	Y-=1 → Y= Y-1 will give Y=1
*=	**Multiply Equal operator** will return the product of right operand and left operand.	X *= Y → X = X * Y, X = 10

/=	**Division Equal operator** will divide right operand by left operand and return the quotient.	X /= Y → X = X / Y, X = 2.5
%=	Modulus Equal to operator will divide right operand by left operand and return the mod (Remainder).	X %= Y is similar to X = X % Y, now X is 1

<u>Examples:</u>

Rewrite the equevelment statmentes for the following expressions anf find the results, assume X=2, Y=3, Z=4, V= 12, C=8.

Example	Equivalent Statement	Result
X += 5	X = X + 5	X ← 7
Y -= 8	Y = Y - 8	Y ← -5
Z *= 5	Z = Z * 5	Z ←
V /= 4		V ←
C %= 3		C ←

<u>Relational Operator:</u>

Relational operators are used for checking conditions whether the given condition is true or false. If the condition is true, it will return non-zero value, if the condition is false, it will return 0.

Suppose we have,

int X = 5, Y = 2;

Operator	Name	Description	Example
>	Greater than	Check whether the left operand is greater than right operand or not.	(X > Y) will return true
<	Smaller than	Check whether the left operand is smaller than right operand or not.	(X < Y) will return false
>=	Greater than or Equal to	Check whether the left operand is greater or equal to right operand or not.	(X >= Y) will return true

<=	Smaller than or Equal to	Check whether the left operand is smaller or equal to right operand or not.	(X <= Y) will return false
==	Equal to	Check whether the both operands are equal or not.	(X == Y) will return false
!=	Not Equal to	Check whether the both operands are equal or not.	(X != Y) will return true

Operator	Name	Example
==	**Equality**	**5 == 5 // gives 1**
!=	**Inequality**	**5 != 5 // gives 0**
<	**Less Than**	**5 < 5.5 // gives 1**
<=	**Less Than or Equal**	**5 <= 5 // gives 1**
>	**Greater Than**	**5 > 5.5 // gives 0**
>=	**Greater Than or Equal**	**6.3 >= 5 // gives 1**

Logical Operators

Logical operators are used in situation when we have more then one condition in a single if statement.

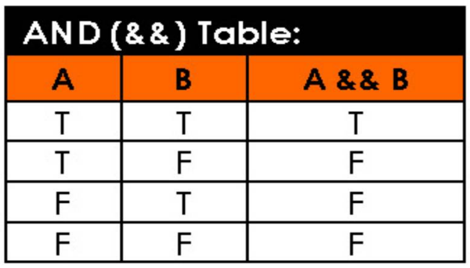
Suppose we have,

int X = 5, Y = 2;

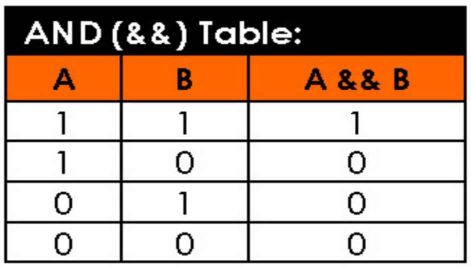
Operator	**Name**	**Description**	**Example**

&&	AND	Return true if all conditions are true, return false if any of the condition is false.	if(X > Y && Y < X) will return true
||	OR	Return false if all conditions are false, return true if any of the condition is true.	if(X > Y || X < Y) will return true
!	NOT	Return true if condition if false, return false if condition is true.	if(!(X>y)) will return false

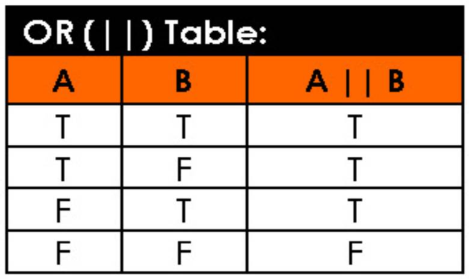
Operator	**Name**	**Example**
&&	**Logical And**	**5 < 6 && 6 < 6 // gives 0**
||	**Logical Or**	**5 < 6 || 6 < 5 // gives 1**
!	**Logical Negation (Not)**	**!(5 == 5) // gives 0**


AND (&&) Table:

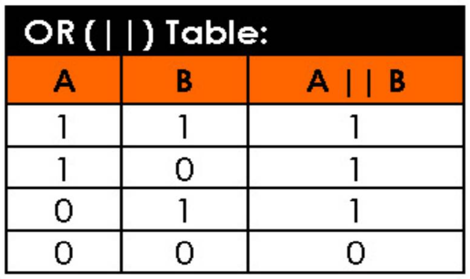
A	B	A && B
T	T	T
T	F	F
F	T	F
F	F	F


AND (&&) Table:

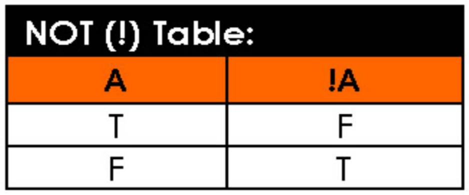
A	B	A && B
1	1	1
1	0	0
0	1	0
0	0	0


OR (| |) Table:

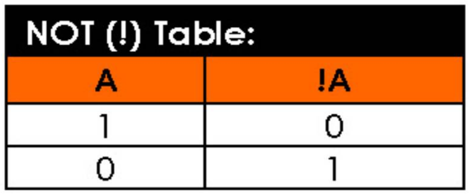
A	B	A | | B
T	T	T
T	F	T
F	T	T
F	F	F


OR (| |) Table:

A	B	A | | B
1	1	1
1	0	1
0	1	1
0	0	0


NOT (!) Table:

A	!A
T	F
F	T


NOT (!) Table:

A	!A
1	0
0	1

Examples: **The following example to understand all the arithmetic operators available in C++.**

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

main()
{
int a = 21;
int b = 10;
int c ;
c = a + b;
cout << "Line 1 - Value of c is :" << c << endl ;

c = a - b;
cout << "Line 2 - Value of c is :" << c << endl ;

c = a * b;
cout << "Line 3 - Value of c is :" << c << endl ;

c = a / b;
cout << "Line 4 - Value of c is :" << c << endl ;

c = a % b;
cout << "Line 5 - Value of c is :" << c << endl ;

c = a++;
cout << "Line 6 - Value of c is :" << c << endl ;

c = a--;
cout << "Line 7 - Value of c is :" << c << endl ;

return 0;
```

```
Line 1 - Value of c is :31
Line 2 - Value of c is :11
Line 3 - Value of c is :210
Line 4 - Value of c is :2
Line 5 - Value of c is :1
Line 6 - Value of c is :21
Line 7 - Value of c is :22
```

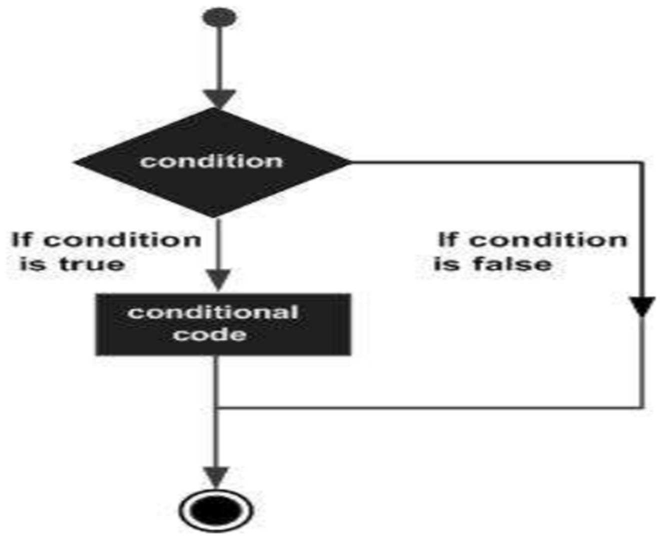
Q/ What's Output:

```
#include<iostream>
using namespace std;
int main()
{ int x,y,z;
x=y=z=0;
x=++y + ++z;
cout<<x<<y<<z<<endl;
x=++y - --z;
cout<<x<<y<<z<<endl;
return 0;
}
```

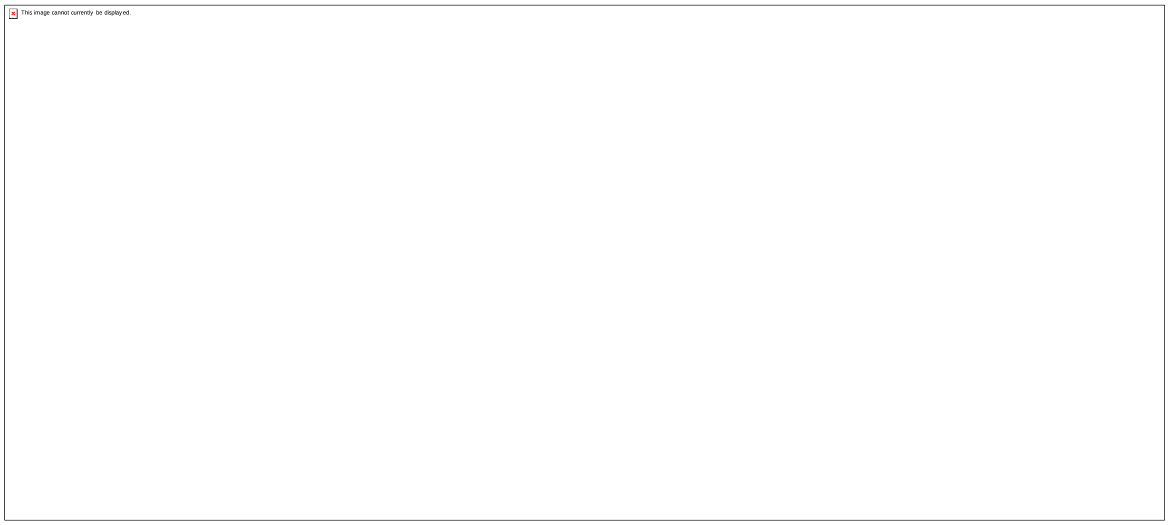
DECISION-MAKING STATEMENTS

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.

Following is the general from of a typical decision making structure found in most of the programming languages:



C++ programming language provides following types of decision making
statements.

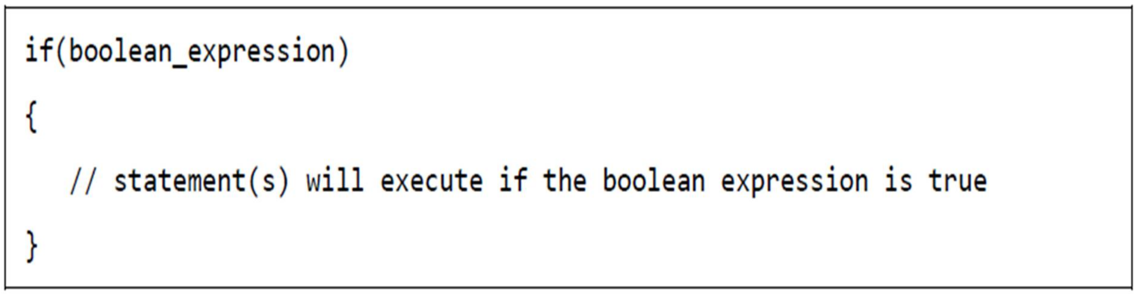


If Statement

if statement consists of a boolean expression followed by one or more statements.

Syntax

The syntax of an if statement in C++ is:



```
if(boolean_expression)
{
   // statement(s) will execute if the boolean expression is true
}
```

If the boolean expression evaluates to **true**, then the block of code inside the if statement will be executed. If boolean expression evaluates to **false**, then the first set of code after the end of the if statement (after the closing curly brace) will be executed.

Example:

Write C++ program to read a given integer value from keyboard and print the value if it is positive.

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

int main()
{
int a;

cout << "Input integer value a :";

cin >>a;

if (a>0)
   cout<<"a is positive number" << endl;

cout << "the value of a is :"<< a;

return 0;
}
```

The output for the above program is : **the input value**

```
Input integer value a : 10

a is positive number
the value of a is: 10
```

if...else Statement

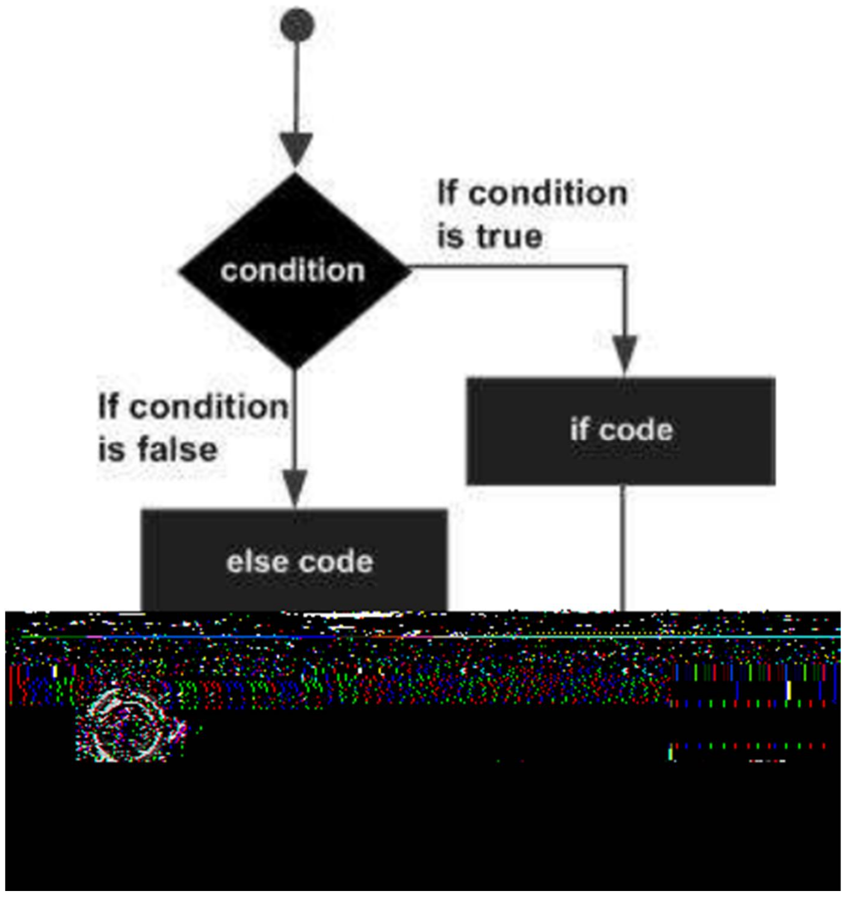
if statement can be followed by an optional **else** statement, which executes
when the boolean expression is **false**.

Syntax

The syntax of an if...else statement in C++ is:

```
if(boolean_expression)
{
   // statement(s) will execute if the boolean expression is true
}
else
{
  // statement(s) will execute if the boolean expression is false
}
```

If the boolean expression evaluates to **true**, then the **if block** of code will be
executed, otherwise **else block** of code will be executed.



Example:

Write C++ program to read a given integer value from keyboard and print the value if it is positive otherwise print it is negative.

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

int main()
{
int a;

cout << "Input integer value a :";

cin >>a;

if (a>0)
   cout<<"a is positive number" << a;

else
   cout <<"a is negative number"<< a;

return 0;
}
```

Ex/Write C++ program to read a given integer value from keyboard and check if
the value is even or odd .

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

int main()
{
int a;
cout << "Input integer value a :";
cin >>a;
if (a % 2 == 0)
   cout<<"a is even number" << a;
else
   cout <<"a is odd number"<< a;
return 0;
```

Example

Write C++ program to calculate Z value according to the following equations:

$$Z = \begin{cases} X + 10 & if\ X > 0 \\ 2X + 50 & if\ X < 0 \end{cases}$$

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

int main()
{
int X, Z;
cout << "Input integer value X :";

cin >>X;

if (X  > 0)
  {
      Z=X+10;
      cout<<" Z value is:" << Z;
  }
else
  {
      Z= 2*X+50;
      cout <<"Z value is :"<< Z;
  }
return 0;
```

Q //write a program to enter a character and determine the character is digit, small or capital letter?

```
#include <iostream>
using namespace std;
int main()
 { char letter;
 cout<<"Enter a letter:"; cin >> letter;
if(letter >= '0'&& letter <= '9') cout<<”entered a digit.";
 else
 if(letter >= 'a' && letter <= 'z') cout<<”entered a small letter.";
 else
  if(letter >= 'A' && letter <= 'Z') cout<<”entered a capital
letter.";
  else
cout<<"You entered a special letter.";
return 0;
}
```

اكتب برنامج لمحاكاة عمل الحاسبة اليدوية

Q / /write program to simulation a simple hand Calculator work?

```
#include<iostream>
using namespace std;
int main()
{
int x,y; char sign;
cin>>x >>sign >>y;
if (sign== '+')
        cout<<x + y;
else if (sign == '-')
        cout<< x - y;
else if (sign == '*')
        cout<< x * y;
else if (sign == '/')
        cout<< x / y;
else
cout << "ERROR" ;
return 0;
}
```

س/اكتب برنامج لادخال رقم بين (1-7) ، وطباعة اسم اليوم المقابل من ايام الاسبوع؟

```
#include<iostream>
using namespace std;
int main()
```

```
{ int d; cout<<"Enter Number (1-7): "; cin>>d;
  if (d==1) cout<<"Sun.";
  else if (d==2) cout<<"Mun.";
  else if (d==3) cout<<"Tues.";
  else if (d==4) cout<<"Wen.";
  else if (d==5) cout<<"Thri.";
  else if (d==6) cout<<"Fri.";
  else if (d==7) cout<<"Sat.";
  else cout<<"number out of range!!!";
return 0;
}
```

The Switch Statement :

The switch statement provides a way of choosing between a set of alternatives, based on the value of an expression. The general form of the switch statement is:

```
switch (expression) {
case constant 1: statements;
...
...
case constant n: statements;
default: statements;
}
```

س/اكتب برنامج لادخال رقم بين (1-7) ، وطباعة اسم اليوم المقابل من ايام الاسبوع؟

Q // Write Program (W.P.) to read number between (1-7) and print name of day?

```
#include<iostream>
using namespace std;
int main()
{ int d;
cout<<"Enter Number (1-7): "; cin>>d;
switch (d)
        {
        case 1: cout<<"Sun."; break;
        case 2: cout<<"Mun.";  break;
        case 3: cout<<"Tues."; break;
        case 4: cout<<"Wen.";  break;
        case 5: cout<<"Thri."; break;
        case 6: cout<<"Fri.";  break;
        case 7: cout<<"Sat.";  break;
        default: cout<<"number out of range!!!";
        } return 0;
}
```

اكتب برنامج لمحاكاة عمل الحاسبة اليدوية

Q//write program to simulation a simple hand Calculator work? (using switch)

```
#include<iostream>
using namespace std;
int main()
{ float a,b; char sign;
cin>>a>>sign>>b;
switch (sign)
        {
        case '+': cout<<"="<<a+b <<endl ; break;
        case '-': cout<<"="<<a-b <<endl ; break;
        case '*': cout<<"="<<a*b <<endl ; break;
        case '/': cout<<"="<<a/b <<endl ; break;
        default:cout<<"ERROR";
        }
return 0;
}
```

and we can use nested switch statement!!

switch (expression$_1$)

{

```
 case constant 1: switch (expression2)
 :
 :
 case constant n:

 default: statements;
}
```

س/اكتب برنامج لقراءة اول حرفيين من بداية اليوم وطباعة اسم اليوم؟

Q5/ /Write Program (W.P.) to read first and second letter from names day starting and print name of day?

```
#include<iostream>
using namespace std;
int main()
{
char t1; char t2;
cout<<"Enter first charater: "; cin>>t1;
switch (t1)
        {
        case 's': {cout<<"Enter second charater: ";
                                                    cin>>t2;
                                  switch(t2)
                                  { case 'u':   cout<<"Sun."; break;
                                    case 'a': cout<<"Sat."; break;
                                    default: cout<<"ERROR !!!"; break;
                                    }break; }
```

H.W: Complete the previous program.

س/اكتب برنامج لقراءة 3 ارقام صحيحة وايجاد الاكبر بينهم؟

Q7/write a program to input 3 integers and determine which of them is biggest? (Second Method)

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z; int larg;
cin>>x>>y>>z;
larg=x;
if (y>larg) larg=y;
if (z>larg) larg=z;
cout<<"larg No.="<<larg;
return 0;
}
```

//find smallest No.?

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z; int small;
cin>>x>>y>>z;
small=x;
if (y<small) small=y;
if (z<small) small=z;
cout<<small;
return 0;
}
```

س/اكتب برنامج لقراءة اول حرفيين من بداية اليوم وطباعة اسم اليوم؟

Q5//Write Program (W.P.) to read first and second letter from names day starting and print name of day?

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

```
char t1,t2;
cout<<"Enter first charater: "; cin>>t1;
switch (t1)
      {
      case 's': {cout<<"Enter second charater: ";
                  cin>>t2;
                  switch(t2)
                  { case 'u':  cout<<"Sun."; break;
                    case 'a': cout<<"Sat."; break;
                    default: cout<<"ERROR !!!"; break;
                    }break; }
      case 'm': cout<<"Mun."; break;
      case 't': {cout<<"Enter second charater: ";
      cin>>t2;
                   switch(t2)
                   { case 'u': cout<<"Tues."; break;
                  case 'h': cout<<"Thri."; break;
                   default: cout<<"ERROR !!!";
                   } break; }
      case 'w': cout<<"wen."; break;
      case 'f': cout<<"Fri."; break;
      default: cout<<"ERROR!!!";
      }
 return 0; }
```

```
#include<iostream>
using namespace std;
int main()
{ int m;
cout<<"enter No. month: "; cin>>m;
switch (m){
case 1:
case 3:
case 5:
case 7:
```

```
case 8:
case 10:
case 12: cout<<"31 days"; break;
case 2:cout<<"28 days"; break;
case 4:
case 6:
case 9:
case 11: cout<<"30 days"; break;
default:cout<<"No. out of range 1_12";
} return 0; }
```

Ex//Enter month Number from (1-12) then display count month days?

س/اكتب برنامج لايجاد قيمة y من المعادلة التالية؟

Q3// Write Program (W.P.) to find Y value , when

$$Y = \sqrt{x^2 + z^2}$$

```
#include<iostream>
#include<math.h>
using namespace std;
int main()
{
int x,z; double y;
cout<<"Enter x value: ";  cin>>x;
cout<<"Enter z value: ";  cin>>z;
        y=sqrt(x*x+z*z);
                cout<<"Y="<<y;
return 0;
}
```

س/اكتب برنامج لايجاد قيمة y من المعادلة التالية؟

Q3// Write Program (W.P.) to find Y value , when

$$Y = \sqrt{X^4} + 5 * X + 3 \qquad IF\ x\ IS\ EVEN$$

$$Y = \sqrt{X^3} + 2 * X + 5 \qquad IF\ x\ IS\ ODD$$

```
  #include<iostream>
#include<math.h>
using namespace std;
int main()
{
int x; double y;
cin>>x;
if (x%2==0)
        y=sqrt(pow(x,4)+5*x+3);
        else
        y=sqrt(pow(x,3)+2*x+5);
cout<<"Y="<<y;
return 0;}
```

Q0//what is the output of this program:

```
#include<iostream>
using namespace std;
int main()
{
int x,y,z;
x=y=z=0;
x=++y + ++z;
cout<<x<<y<<z<<endl;
x=y++ + z++;
cout<<x<<y<<z<<endl;
x=++y + z++;
cout<<x<<y<<z<<endl;
x=y-- + --z;
cout<<x<<y<<z<<endl;
return 0;
}
```

Output screen

```
--------------------
211
```

222
533
522

Loops

- A loop statement allows us to execute a statement or group of statements multiple times and following is the general from of a loop statement in most of the programming languages

Loop Types:

C++ programming language provides the following type of loops to handle
looping requirements.

Loop Type	Descryption
for loop	The initialization, condition and increment/decrement all put together. Initialization will be done once at the beginning of loop. Then, the condition is checked by the compiler. If the condition is false, for loop is terminated. But, if condition is true then, the statements are executed until condition is false.
while Loop	Repeats a statement or group of statements while given condition is true. It tests the condition before executing the loop body.
do...while loop	Like a 'while' statement, except that it tests the condition at the end of the loop body.
nested loops	You can use one or more loop inside any another 'while', 'for' or 'do..while' loop.

For Loop: A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax of For Loop

```
        for ( initialization ;  condition; increment )
        {
            Statements ;
        }
```

Here is the flow of control in a for loop:

- The (initialization step is executed first, and only once. This step allows you to declare and initialize any loop control variables.
- Next, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and flow of control jumps to the next statement just after the for loop.
- After the body of the for loop executes, the flow of control jumps back up to the **increment** statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.
- After the condition becomes false, the for loop terminates.

Example:

```
#include<iostream>
```

using namespace std;

```
    void main()
    {
        int a, num;
        cout << "Enter any number : ";
        cin >> num;
```

```
            for (a=1;a<=num;a++)
             cout << "\nHello...!!";      }
```

The output:

```
Enter any number : 5
           Hello...!!
           Hello...!!
           Hello...!!
           Hello...!!
           Hello...!!
```

-For Statement:

for(exp1 ; exp2 ; exp3)

When:

exp1: First counter

exp2: End counter

exp3: Increment or Decrement Counter

Example:

for(i=1;i<=10;i++) cout<<i; // = 123…910

for(i=10;i>=1;i--) cout<<i; // = 1098…321

for(i=2;i<=10;i+=2) cout<<i; // = 246810

ماهو الفرق بين السؤالين التاليين:

حساب مجموع الاعداد بين (1-10)؟

الفرق سيكون بين المعلوم والمجهول
حساب مجموع الاعداد بين (1-10)؟ (معلوم)

for(i=1;i<=10;i++)
s+=i;

حساب مجموع 10 اعداد؟ (مجهول)

```
for(i=1;i<=10;i++)
{ cin>>x;
  s+=x; }
```

حساب مجموع 10 اعداد؟

EX// W.P. to read 10 integer number and find biggest?

```
#include<iostream>
using namespace std;
int main()
{
int i,x,larg=0;
cin>>x;
larg=x;
for(i=1;i<=9;i++)
{
cin>>x;
if(x>larg) larg=x;
}
cout<<"larg="<<larg;
 return 0;
}
```

EX// W.P. to find factorial x! ?

```
#include<iostream>
using namespace std;
int main()
{
int i,x,f=1;
cin>>x;
for(i=1;i<=x;i++)
f*=i;
cout<<"factorial x="<<f;
return 0;
}
```

EX// W.P. to find power x^y ?
(without using pow function)

```
#include<iostream>
using namespace std;
int main()
{
int i,x,y,p=1;
cin>>x>>y;
for(i=1;i<=y;i++)
p*=x;
cout<<"power="<<p;
return 0;
}
```

While Loop

While loop is also called entry control loop because, in while loop, compiler will 1st check the condition, whether it is true or false, if condition is true then execute the statements.

Syntax of While Loop

```
while ( condition )
{
    Statmentes;
   --- - -- - -- -- - -
}
```

- Here, **statement(s)** may be a single statement or a block of statements.

- The **condition** may be any expression, and true is any non-zero value. The loop iterates while the condition is true. When the condition becomes false, program control passes to the line immediately following the loop.

EX/ print even numbers between (4-100)?

using (For, while, do—while) statement

```
for (i=4;i<=100;i+=2)

cout<<i;     // 4 6 8 ... 98 100 by for
```

--

```
i=4;
while (i<=100)
{ cout<<i;
i+=2; }     // 4 6 8 ... 98 100 by while
```

```
i=4;
do
{ cout<<i;
  i+=2;
 } // 4 6 8 ... 98 100 by do-while
while (i<=100);
```

س/اكتب برنامج لحساب مجموع الارقام الصحيحة التي تقبل القسمة على 4 و 5 من 10 ارقام مختلفة؟

Q1//write a program to compute the sum of the integers that divisible by 4 and 5 from 10 different integers?

س/اكتب برنامج لحساب مجموع الارقام الموجبة لقائمة من الارقام تنتهي بالرقم صفر؟

Q2// Write a program to calculate the sum of the positive numbers to the list of numbers ending the number zero?

الحلقات او التكرارات- Loops OR Iterations :

Removing all the expressions gives us an infinite loop. This loop's condition is assumed to be always true:

for (; ;) // infinity loop

الحلقات او التكرارات- Loops OR Iterations :

Because loops are statements, they can appear inside other loops. In other words, loops can be nested. For example,

```
for (int i = 1; i <= 3; ++i)
for (int j = 1; j <= 3; ++j)
cout << "(" << i << "," << j << ")\n";
```

```
Output :
-----------
(1,1)
(1,2)
(1,3)
(2,1)
(2,2)
(2,3)
(3,1)
(3,2)
(3,3)
```

الحلقات او التكرارات- Loops OR Iterations :

Q//write a program used nested loop to calculate "X" ,where:- X= 1!+ 3!+ 5!+ 7!+9!

```
#include<iostream>
using namespace std;
 int main()
 { int i,j,f=1,x=0;
 for(i=1;i<=9;i+=2)
 { f=1;
 for(j=1;j<=i;j++)
 f*=j;
 cout<<"factorial " <<i<<"= "<<f<<endl;
 x+=f; }
  cout<<"Sum ="<<x<<endl;
return 0;   }
```

الحلقات او التكرارات- Loops OR Iterations :

Q//what's output:

```
#include<iostream>
using namespace std;
 int main()
 { int i,j;
 for(i=1;i<=4;i++)
 {
 for(j=1;j<=4;j++)
cout<<"*";
cout<<endl;
  }
return 0;
}
```

```
Output :
-----------
****
****
****
****
```

Example

```
#include <iostream>
using namespace std;
int main ()
{
// Local variable declaration:
int a = 10;
// while loop execution
while( a < 20 )
{
```

The output:

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
.
.
value of a: 19
```

يقوم هذا البرنامج بطباعة قيمة المتغير a طالما الشرط (a<20) متحقق او TRUE

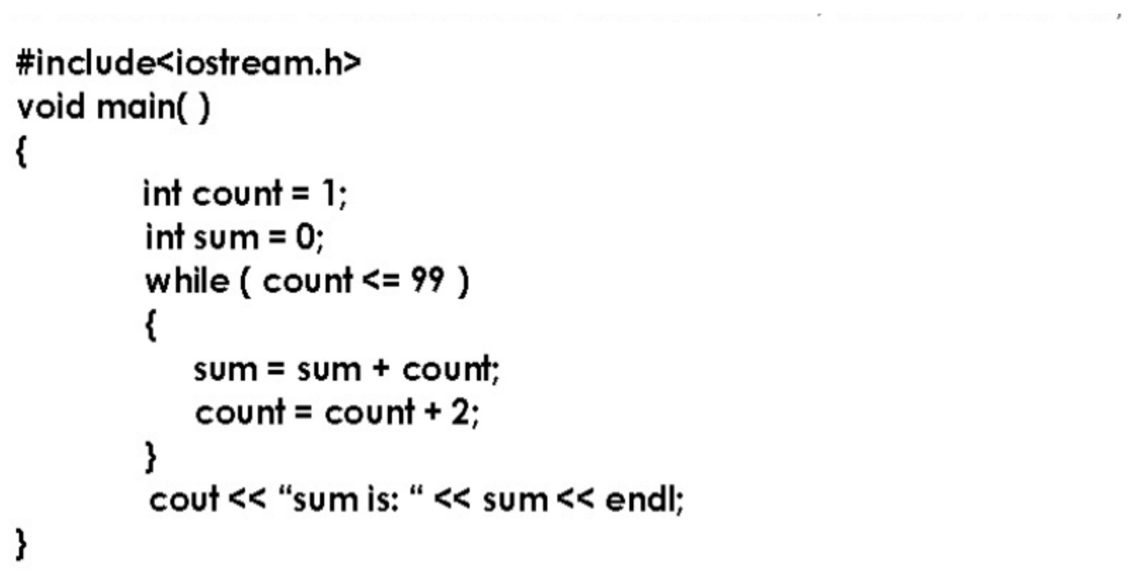
Homework: trace the following c++ cods and Find the final output for these cods :

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

int main()
{
    int n, sum = 0;
    cout << "Enter a positive integer: ";
    cin >> n;

    for (int i = 1; i <= n; ++i) {
        sum += i;
    }
```

```
   cout << "Sum = " << sum;
}
```

Exercises

1. Write C++ program to find the summation of the following series: Sum= 1+3+5+7+ . . . . +99



```
#include<iostream.h>
void main( )
{
           int count = 1;
           int sum = 0;
           while ( count <= 99 )
           {
               sum = sum + count;
               count = count + 2;
           }
           cout << "sum is: " << sum << endl;
}
```

2. Write C++ program to read 10 integer numbers, and find the sum of the positive numbers only?

3. **Write C++ program to find the summation of the following series :**

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2$$

4. Write C++ program to find the summation of students marks, and its average, assume the student have 8 marks.

The break Statement

A *break* statement may appear inside a loop (while, do, or for) or a switch statement. It causes a jump out of these constructs, and hence terminates them.

EX// read positive No. and terminate by zero?

```
for( ; ; )  //infinity loop
{   cin >> num;
if (num < 0) continue;
If (num==0) break;
 // process num here...
}
```