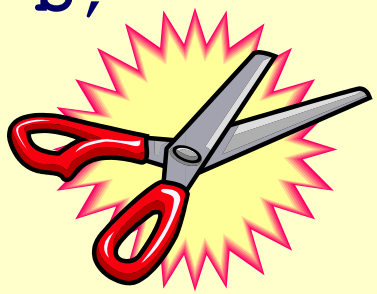
Programming 1

while – For – Do While Loops

الاسنادات المختصرة

- C++ لها بعض العمليات المختصرة

	shortcut	same as
*=	`a *= b;`	`a = a*b;`
/=	`a /= b;`	`a = a/b;`
+=	`a += b;`	`a = a+b;`
-=	`a -= b;`	`a = a-b;`
%=	`a %= b;`	`a = a%b;`



أمثلة:

```
int i = 3;
i += 4;                    // i = i + 4
cout << i << endl;         // i is now 7

double a = 3.2;
a *= 2.0;                  // a = a * 2.0
cout << a << endl;         // a is now 6.4

int change = 1265;
change %= 100;             // change = change % 100
cout << change << endl; // change is now 65
```

الحلقات (البنية التكرارية) Loops (Iterative Constructs)

تسمح الحلقات بتنفيذ المقطع البرمجي (الكود) عدة مرات.

يوجد ٣ أنواع لبنى التكرار:

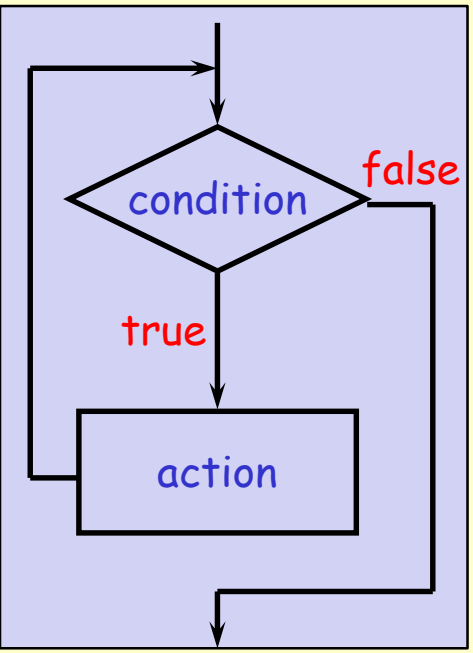
Three constructs

- while statement
- for statement
- do-while statement

The while Statement

- الشكل العام:

```
while (condition)
    action
```



- كيفية عملها:
  - إذا كانت الشرط(condition) true ، فقم بتنفيذ الإجراء action
  - كرر هذه العملية حتى يتم يصبح الشرط(condition) false
- الإجراء action إما عبارة واحدة أو مجموعة من العبارات داخل الأقواس

```
while (it's raining){
    <keep the umbrella up>
}
```

مثال

```
#include <iostream>
using namespace std;
int main()
{
    char answer;
            cout << "Do you want to play the game: (y/n) ";
            cin >> answer;

            while ( answer == 'y')
            {
                    cout << "Play game ...." << endl;
                    cout << "Do you want to play again: (y/n) ";
                    cin >> answer;
            }
            return 0;
}
```

مثال: Enter a number between 1-- 4

```
#include <iostream>
using namespace std;
int main()
{
   int answer, counter = 1;
            cout << "Please enter a number between 1--4:";
            cin >> answer;
            while ( answer < 1 || answer > 4)
            {           cout << "Please enter a number between 1--4:";
                        cin >> answer;
                        counter +=1;  }
            cout << "Thank you for your splendid choice after "
                        << counter << " try !" << endl;
            return 0;
}
```

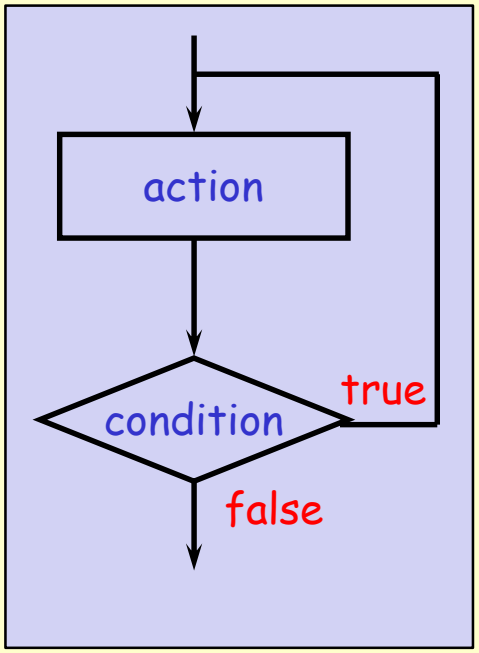
مثال : N! العاملي (While)

n! = 1*2*3*...*n

```
int number, factorial, counter;
cout << "Enter a positive integer:";
cin >> number;
factorial = 1;
counter = 1;
  while(counter <= number){
            factorial *= counter;
            counter += 1; //counter = counter + 1;
            }
cout << "The factorial of " << number << " is " << factorial << endl;
```

The do-while Statement

- الشكل العام:



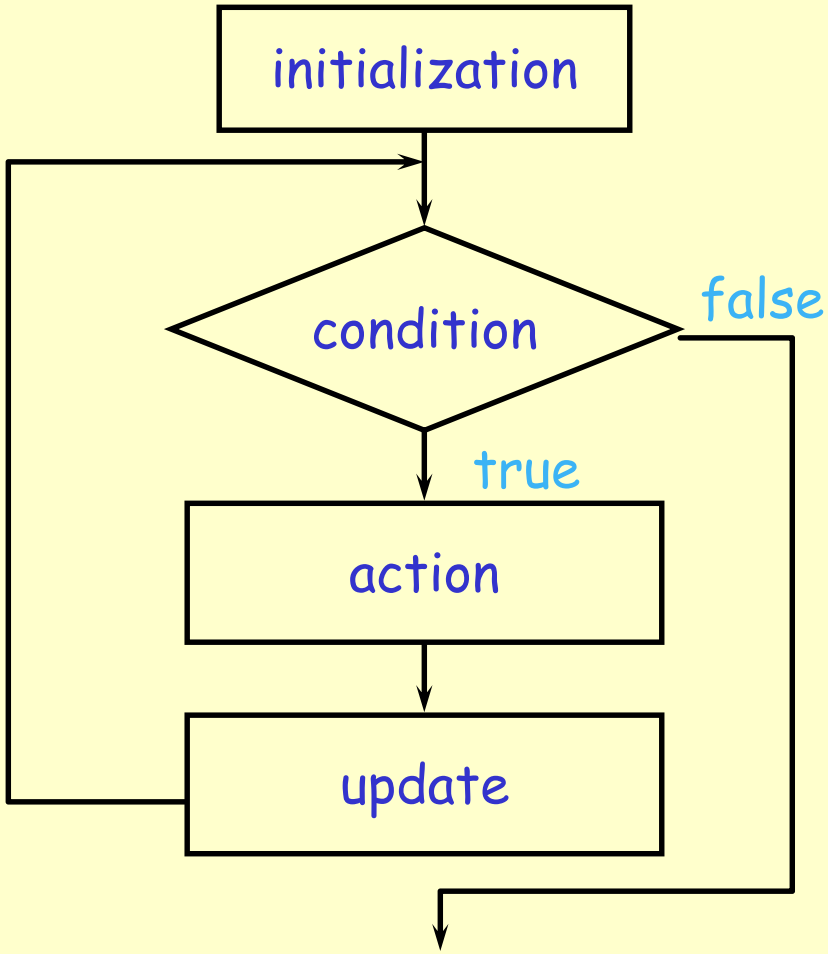

```
do action
   while (condition)
```

- كيفية عملها:
  - نفذ الإجراء action
  - إذا كانت الشرط(condition) true ، فقم بتنفيذ الإجراء action مرة ثانية
  - كرر هذه العملية حتى يتم يصبح الشرط(condition) false
- الإجراء action إما عبارة واحدة أو مجموعة من العبارات داخل الأقواس

مثال : N! العاملي (Do While)

```
int number, factorial, counter;
      cout << "Enter a positive integer:";
      cin >> number;
      factorial = 1;
      counter = 1;
      do{
            factorial *= counter;
            counter++;
      }while(counter <= number);
cout << "The factorial of " << number
     << " is " << factorial << endl;
```

The for Statement



The do-while Statement

- الشكل العام:

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

- كيفية عملها:
  - نفذ تعليمة التهيئة initialization
  - طالما الشرط(condition) true
    - نفذ الإجراء action
    - نفذ التعديل update

- مثال:

```
int i;
for(i=1; i<=20; i++)
cout << "i is " << i << endl;
```

مثال : N! العاملي (For)

```
int number, factorial, n;
        cout << "Enter positive integer:";
        cin >> number;
        factorial = 1;
        for(n=1; n<=number; n++)
                factorial *= n;
        cout << "The factorial of " << number
                << " is " << factorial << endl;
```

Break & Continue

- يتم استخدام تعليمة **break** عندما نريد إنهاء حلقة قبل أن تنتهي بطريقة طبيعية.
- تعليمة **continue** تكسر تكرارًا واحدًا (في الحلقة) ، في حالة حدوث شرط محدد ، وتستمر مع التكرار التالي في الحلقة.

```
int value=0;          // input value
int max=0;            // maximum value
while(true)           // infinite loop!!!
{
cout << "Enter a positive integer " << "(-1 to stop):";
cin >> value;
if(value > max)
   max = value;
  if(value== -1) break;
}
  cout << "The maximum value found is " << max << endl;
```

أمثلة: Continue & Break

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    break;
  }
  cout << i << "\n";
}
```

```
0
1
2
3
```

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    continue;
  }
  cout << i << "\n";
}
```

```
0
1
2
3
5
6
7
8
9
```

أمثلة: Continue & Break

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

int main() {
  int i = 0;
  while (i < 10) {
    cout << i << "\n";
    i++;
    if (i == 4) {
      break;
    }
  }
  return 0;
}
```

```
0
1
2
3
```

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

int main() {
  int i = 0;
  while (i < 10) {
    if (i == 4) {
      i++;
      continue;
    }
    cout << i << "\n";
    i++;
  }
  return 0;
}
```

```
0
1
2
3
5
6
7
8
9
```

الزيادة والنقصان(Increment and Decrement)

لدى C ++ عمليات خاصة للزيادة (++) للنقصان (--) بمقدار عدد صحيح واحد. تعمل العملية ++ على النحو التالي:

- ++k يزيد قيمة k بمقدار واحد ويتم استخدام القيمة المتزايدة في التعبير.
- k++ يستخدم القيمة الأولية لـ k في التعبير والزيادات بعد ذلك.

```
int k = 1, j;
 j = k++; // First: j is assigned to 1
          //Second: k is increased by 1, becomes 2
     k = 3;
     cout << k++;
     //First:3 printed on the screen
     // Second:k is increased by 1, becomes 4
```

```
int k = 1, j;
j =++k; //First: k increased by 1, becomes 2,
        // Second: j is assigned to 2
k = 3;
cout << ++k;
 // First: k is increased by 1, becomes 4,
 // Second: 4 outputs to the screen
```

الزيادة والنقصان(Increment and Decrement)

تعمل العملية -- على النحو التالي:

❖ --k ينقص قيمة k بمقدار واحد ويتم استخدام القيمة المتناقصة في التعبير.

❖ يستخدم k-- القيمة الأولية لـ k في التعبير والنقصان بعد ذلك.

```
int k = 1, j;
j = k--; // first: j is assigned to 1
         // second: k decreased by 1, becomes 0,
k = 3;
cout << k--;
//First: 3 output to the screen
//Second: k is decreased by 1, becomes 2
```

```
int k = 1, j;
j =--k; // First: k decreased by 1, become 0,
        // Second: j is assigned to 0
k = 3;
cout << --k;
// First: k is decreased by 1, becomes 2, //
Second: 2 is output to the screen
```

مثال:

```
int k = 4;

cout << k << endl;
cout << ++k << endl;  // k=k+1 : k is 5
cout << k++ << endl;  // k=k+1 : k is 6

int i = k++;                  // i is 6, k is 7
cout << i << " " << k << endl;

int j = ++k;          // j is 8, k is 8
cout << j << " " << k << endl;
```

مثال

```
int k = 4;

cout << k << endl;
cout << --k << endl;  // k=k-1 : k is 3
cout << k-- << endl;  // k=k-1 : k is 2

int i = k--;                  // i is 2, k is 1
cout << i << " " << k << endl;

int j = --k;          // j is 0, k is 0
cout << j << " " << k << endl;
```