

النوت 20 صفحة سحب كمبيوتر:  
ملونة: دينار كويتي واحد  
أبيض وأسود: ربع دينار فقط

# C++ Programming

## Week 2

### If & While Statements

### (الشرح والأفكار الرئيسية)

#### نوت السي ++

يحتوي على شرح المواضيع  
وأمثلة للأفكار الرئيسية من  
هوموركات وامتحانات سابقة.

#### نوت السي ++

تتكون النوت من عشرة  
أجزاء بمعدل جزء كل أسبوع.

#### لماذا لا تقتنى الأحدث؟

النوت يتم تنقيحها وتجديدها نهاية  
كل أسبوع، راجع [eng-hs.net](http://eng-hs.net)  
للتأكد من شرائك الإصدار الأحدث.

#### نوت السي ++

يتوفر على الموقع ملفات الشرح  
والأفكار الرئيسية، وملفات  
لتمارين وأمثلة إضافية.

فرق كبير بين العزم على  
النجاح، والعزم على عدم الفشل.

## Control Statements

### جمل شرطية Conditional

#### if Statement

تصلح لأي نوع شرط،  
ويمكن الاستغناء بها عن  
استخدام جملة (switch).

#### switch Statement

تصلح في حالات خاصة  
حيث تكون ذات شكل  
مرتب أكثر من جملة (if).

### جمل تكرارية Repetition

#### while loop

عدد محدد أو غير محدد  
من مرات التنفيذ

#### do while loop

#### for loop

عدد محدد  
فقط

\* جملة الـ **while** تصلح لأي نوع تكرار ويمكن الاستغناء بها عن **do**، **for**.

\* استخدام **do** أفضل عندما يكون من الضروري تنفيذ الـ **loop** مرة على الأقل قبل السؤال عن شرطها.

\* استخدام **for** أسهل من استخدام **while** عندما يكون عدد مرات التكرار محدداً ومعروفاً مسبقاً.

## More Operators (++ , -- )

\* **(Increment Operator ++)** يضيف واحد إلى قيمة المتغير سواء تم كتابته قبل أو بعد المتغير والفرق هو ترتيب تنفيذه مقارنة مع باقي الرموز.

x = 5;

y = ++x + 7;

cout << x << endl << y;

تم تنفيذ (++) قبل (+) 6  
13

x = 5;

y = x++ + 7;

cout << x << endl << y;

تم تنفيذ (+) قبل (++) 6  
12

\* في حال إضافة قيمة إلى مخزن به قيمة، أو طرح قيمة من المخزن، أو ضرب أو قسمة المخزن على قيمة يمكن اختصار العملية الحسابية حسب الأمثلة التالية:

y = 7;

sum = 13;

sum = sum + y;

cout << sum; ← نفس النتيجة → cout << sum;

y = 7;

sum = 13;

sum += y;

لا تنافس إلا على القمة.

أشكال مختلفة لجملة ( if )

	true	false	
if ( condition ) statement1; else { statement2; statement3; }	succeed	end-year	if ( mark >= 60 ) cout << "succeed\n"; else { cout << "end-"; cout << "year\n"; }

	true	false	
if ( condition ) { statement1; statement2; } statement3;	succeed end-year	year	if ( mark >= 60 ) { cout << "succeed\n"; cout << "end-"; } cout << "year\n";

	true	false	
if ( condition ) statement1; else statement2; statement3;	succeed end	fail end	if ( mark >= 60 ) cout << "succeed\n"; else cout << "fail\n"; cout << "end\n";

	true	false	
if ( condition ) statement1; statement2; statement3;	succeed end-year	end-year	if ( mark >= 60 ) cout << "succeed\n"; cout << "end-"; cout << "year\n";

	cond1 true	cond2 true	both false	both true	
if (condition1) statement1; else if (condition2) statement2; else statement3; statement4;	succeed end	pass end	fail end	succeed end	if ( mark >= 60 ) cout << "succeed\n"; else if ( mark >= 50 ) cout << "pass\n"; else cout << "fail\n"; cout << "end";

كل إنسان يستطيع أن يبدأ  
من جديد، تماماً من جديد.

## حالات خاصة لجملته (if) وملاحظات

\* في حال عدم ذكر أحد علامات المقارنة الست مع شرط جملة الـ **if** (**>**, **>=**, **<**, **<=**, **==**, **!=**) يكون ناتج الشرط (**true**) مهما كانت القيمة أو ناتج العملية الحسابية داخل قوسي الشرط إلا في حالتي (**0, false**)، فإن الناتج يكون (**false**).

<pre>x = 7; if (x) —;</pre>	true	<pre>x = 0; if (x) —;</pre>	false
<pre>x = 7; if (x / 3) —;</pre>	true	<pre>x = 7; if (x % 7) —;</pre>	false
<pre>x = 7; if (3 / 7) —;</pre>	false	<pre>x = 7; if (3 % 7) —;</pre>	true

\* في حال وضع ( ; ) بعد قوسي الشرط يتم تنفيذ ناتج الشرط بغض النظر عن كون الشرط (**true**) أو (**false**).

<pre>if ( شرط ); —;</pre>	يتم تنفيذها	<pre>if ( شرط ); { —; —; —; }</pre>	يتم تنفيذها كلها
---------------------------	-------------	-------------------------------------	------------------

لعله من عجائب الحياة أنك إذا  
رفضت كل ما هو دون مستوى  
القمة، فإنك غالباً ستحصل عليه.

if statement (output)Find the exact output:

void main ( )

{

// output # 1

int a = 5, b = 7, c = 10;

int sum = 0;

if (a &gt; b &amp;&amp; b &lt; c)

if (b &lt; 3)

sum = a;

else

sum = b;

cout &lt;&lt; "Sum = " &lt;&lt; sum &lt;&lt; endl;

Sum = 0

if (a &gt; b &amp;&amp; b &lt; c)

if ( b &lt; 3 )

sum = a;

else

sum = b;

وهمية لبيان الارتباط  
sum = some value;

// output # 2

int i = 5, j = 7, k = 3;

if (i &lt; j || j &lt; k)

{

if (j &gt;= k \* 2 &amp;&amp; j &lt; k)

k = i;

}

else

k = j + k;

cout &lt;&lt; "k = " &lt;&lt; k &lt;&lt; endl;

k = 3

Trickey Outputsهذه نماذج من  
امتحانات سابقة.

// output # 3

a = 5, b = 7, sum = 3;

if (a &lt; b)

{

b = sum;

sum += b - a;

}

else

sum = b; sum += a;

cout &lt;&lt; "Sum = " &lt;&lt; sum &lt;&lt; endl;

}

a	b	sum
5	7	3
	3	1
		6

Sum = 6

هذه شغالة

الشر الذي تسببه لنفسك أضعاف ما  
يسببه لك الآخرون، كفاك تردداً وخوفاً!

if statement (*minimum-of-3 integers*)

**Write a program that reads three integers and prints their minimum.**

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

```
int main ( )
{
```

```
    int x, y, z, min;
```

```
    cout << "Enter 3 integers: ";
```

```
    cin >> x >> y >> z;
```

```
    if (x < y && x < z)
```

```
        min = x;
```

```
    else if (y < z)
```

```
        min = y;
```

```
    else
```

```
        min = z;
```

```
    cout << "Minimum is: " << min << endl;
```

```
    return 0;
```

```
}
```

Enter 3 integers: **5 8 7**

Minimum is: **5**

Enter 3 integers: **7 18 3**

Minimum is: **3**

يمكن حساب min  
أو max بعدة طرق!  
هذه الطريقة هي إحداها.

المصيبة إذا نزلت فهي واحدة، فإن  
جزع صاحبها كانت اثنين.

if statement (even or odd)

Write a program that reads an integer and indicates if it is even or odd.

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

```
int main ( )
{
```

```
    int x;
```

```
    cout << "Enter an integer: ";
```

```
    cin >> x;
```

```
    if (x % 2 == 0)
```

```
        cout << x << " is even\n";
```

```
    else
```

```
        cout << x << " is odd\n";
```

```
    return 0;
```

```
}
```

Enter an integer: 25  
25 is odd

Enter an integer: 376  
376 is even

أي رقم يقبل القسمة على (2) بدون  
باقٍ يكون even، وإلا يكون odd.



أن يغضب المرء يعني أن يثار من  
نفسه لأخطاء ارتكبها الآخرون.

if statement (*positive or negative*)

Write a program that reads a number and indicates if it is positive or negative.

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

```
int main ( )
{
```

```
    double num;
```

```
    cout << "Enter a number: ";
```

```
    cin >> num;
```

```
    if (num > 0)
```

```
        cout << num << " is positive\n";
```

```
    else if (num < 0)
```

```
        cout << num << " is negative\n";
```

```
    return 0;
```

```
}
```

Enter a number: **19.5**  
**19.5** is positive

Enter a number: **-36**  
**-36** is negative

Enter a number: **0**

في حال (0) ماكو output.

تصرف كما لو أنه من  
المستحيل أن تفشل.



if statement (*reverse sign*)

Write a program that reads a number and prints its additive reverse according to the following output samples.

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

```
int main ( )
{
```

```
double num;
```

```
cout << "Enter a number: ";
```

```
cin >> num;
```

```
if (num != 0)
```

```
cout << "Reverse is: " << num * -1 << endl;
```

```
else
```

```
cout << "No reverse\n";
```

```
return 0;
```

```
}
```

Enter a number: **-7.1**

Reverse is: **7.1**

Enter a number: **0**

*No reverse*

Enter a number: **5.2**

Reverse is: **-5.2**

يفضل جمع الحالات المتشابهة أثناء  
كتابة جملة (if) قدر الممكن.

لا تلم نفسك على قرار اتخذته قط، فقد  
كان هو قرارك الأمثل وقت اختياره.

if statement (multiple)

Write a program that reads two integers and indicates if one of them is multiple of the second.

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

```
int main ( )
{
```

```
    int n1, n2;
```

```
    cout << "Enter two integers: ";
```

```
    cin >> n1 >> n2;
```

```
    if (n1 % n2 == 0)
```

```
        cout << n1 << " is multiple of " << n2 << endl;
```

```
    else if (n2 % n1 == 0)
```

```
        cout << n2 << " is multiple of " << n1 << endl;
```

```
    else
```

```
        cout << "no multiples\n";
```

```
    return 0;
```

```
}
```

Enter two integers: **7 21**  
**21** is multiple of **7**

Enter two integers: **18 3**  
**18** is multiple of **3**

Enter two integers: **17 5**  
**no multiples**

أي رقم يكون مضاعف لرقم آخر إذا  
كان باقي ناتج القسمة مساوياً لـ (0).

يموت الجبناء مرات عديدة قبل  
أن يأتي أجلهم، أما الشجعان  
فيذوقون الموت مرة واحدة.

if statement (*ascending of 2 integers*)

Write a program that reads two integers and prints them in an ascending order.

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

```
int main ( )
{
```

```
    int n1, n2;
```

```
    cout << "Enter two integers: ";
```

```
    cin >> n1 >> n2;
```

```
    if ( n1 < n2 )
```

```
        cout << "Ascending is: " << n1 << " " << n2 << endl;
```

```
    else
```

حالة else تشمل الحالتين الأخريين

```
        cout << "Ascending is: " << n2 << " " << n1 << endl;
```

```
    return 0;
```

```
}
```

Enter two integers: **14 2**  
Ascending is: **2 14**

Enter two integers: **7 21**  
Ascending is: **7 21**

Enter two integers: **5 5**  
Ascending is: **5 5**

يجب ترك مسافة أثناء الطباعة  
حتى لا يلتصق الرقمان.

يولد الطفل واثقاً، لكن  
الخوف يأتي بالتعلم.

if statement (*payment*)

Write a program that reads a positive number representing hours worked by a labor and prints his equivalent payment if he gets as follows:

**2.00 KD/hour for first 30 hours.**

**2.50 KD/hour for next 20 hours.**

**3.00 KD/hour for further hours.**

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

```
int main ( )
```

```
{
```

```
    int hours;
```

```
    double pay;
```

```
    cout << "Enter hours: ";
```

```
    cin >> hours;
```

```
    if (hours <= 30)
```

```
        pay = hours * 2;
```

```
    else if (hours <= 50)
```

```
        pay = 30 * 2 + (hours - 30) * 2.5;
```

```
    else
```

```
        pay = 30 * 2 + 20 * 2.5 + (hours - 50) * 3;
```

```
    cout << "Payment is: " << pay << " KD\n";
```

```
    return 0;
```

```
}
```

Enter hours: **25**

Payment is: **50** KD

Enter hours: **35**

Payment is: **72.5** KD

Enter hours: **70**

Payment is: **170** KD

أحد المسائل المشهورة  
للتدريب على (nested If).

يدرك الإنسان أهمية الوقت عندما  
تحين اللحظات الأخيرة لأمر ما،  
أغلبنا ينتظر هذا الوقت كي يندم.

Count-Control while (output)Find the exact output:

```

i = 1;
while (i <= 5)
{
    cout << "We may love C++";
    cout << endl;
    i++;
}
cout << i * 10 << endl;

```

We may love C++  
 We may love C++  
 We may love C++  
 We may love C++  
 We may love C++  
 60

لا تخرج (i) من loop بقيمة (5) بل (6) لأن  
loop (i = 5) تسبب استمرار تنفيذ الـ

Find the exact output:

```

i = 1;
cout << "Enter 5 integers:\n";
sum = 0;
while (i <= 5)
{
    cin >> x;
    if ( x % 2 != 0)
        sum += x;
    i++;
}

```

عناصر  
 أساسية  
 عند قراءة  
 عدد محدد  
 من القيم

i	x	sum = 0
1	7	7 (same)
2	4	7
3	11	18
4	2	18 (same)
5	3	21
6		

Enter 5 integers:  
 7 4 11 2 3  
 Sum of odd numbers: 21

```

cout << "Sum of odd numbers: " << sum << endl;

```

الحاجة أرخص ما تكون  
 حين يستغنى عنها.

## Count-Control while (*count of non-zeros*)

Write a program that reads 10 integers and prints the count of none zeros.

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

Enter 10 integers:

**0 4 0 -10 0 6 8 -3 -6 0**

Count non-zeros is: **6**

```
int main ( )
{
    int i, x, count = 0;

    cout << "Enter 10 integers:" << endl;

    i = 1;

    while ( i <= 10 )
    {
        cin >> x;
        if ( x != 0 )
            count ++;

        i ++;
    }

    cout << "Count non-zeros is: " << count << endl;

    return 0;
}
```

الجميل المطلوب تنفيذها عدة مرات  
هي التي يتم وضعها داخل الـ loop.

لا يمكنك أن تبني سمعة  
على ما ستفعله لاحقاً.

count-control while (average of odds)

Write a program that reads 10 integers and prints the average of only odds.

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

```
int main ()
{
```

```
    int i, x, sum = 0, count = 0;
```

```
    cout << "Enter 10 integers:" << endl;
```

```
    i = 1;
```

```
    while ( i <= 10 )
    {
```

```
        cin >> x;
```

```
        if ( x % 2 != 0 )
```

```
        {
```

```
            sum += x;
```

```
            count ++;
```

```
        }
```

```
        i++;
```

```
    }
```

```
    if ( count == 0 )
```

```
        cout << "No odd numbers" << endl;
```

```
    else
```

```
        cout << "Average of odd numbers is: " << sum * 1.0 / count << endl;
```

```
    return 0;
```

```
}
```

Enter 10 integers:

**17 4 5 10 7 6 8 2 6 9**

Average of odd numbers is: **9.50**

Enter 10 integers:

**8 4 2 10 6 4 8 2 6 4**

No odd numbers

الجميل التي يتم تنفيذها مرة واحدة  
يتم وضعها خارج الـ loop.

مطلوب ناتج كسري

لا يوجد سبب يحبك الناس من أجله  
أفضل من عدم التطلع لما في أيديهم.

## Event-Control while (output)

### Find the exact output:

```
void main ()
```

```
{
    int x, sum = 0;

    cin >> x;
    while (x != -1)
    {
        if (x >= 5)
            sum += x;
        cin >> x;
    }
    cout << "Sum is = " << sum << endl;
}
```

لن يتم قراءة هذه القيمة لأن  
الـ loop تتوقف عند قراءة (-1)

x	sum
4	0
11	11
-1	11

//input is: 4 11 -1 8

Sum is = 11

```
void main ()
```

```
{
    int j = 1;

    while (j <= 10)
    {
        cout << "We hate C++\n";
        j = j - 3;
    }
}
```

j
1
-2
-5
-8

الشرط سيظل متحقق للأبد

We hate C++

We hate C++

We hate C++

.....

.....

.....

(Infinite loop)

```
void main ()
```

```
{
    int i = 0, j = 12;

    while (i <= j);
    {
        i++;
        j--;
    }

    cout << "i = " << i << "\nj = " << j;
}
```

Did you notice (;)?

إذا كان شرط الـ while متحققاً و يوجد بعد  
قوسها (;) ستصبح (infinite loop)

No output

ما رأيتم ظالماً أشبه  
بمظلوم من الحاسد.



## Event-Control while (count of positive odds)

Write a program that keeps reading integers until 0 is entered and prints the count of only positive odd numbers.

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

```
int main ( )
{
```

- keeps reading  
- until (number)  
دلالات على loop غير  
محددة العدد.

Enter integers (0 to end):

**-5 10 -7 6 18 0 61 31**

No positive odd numbers!

Enter integers (0 to end):

**5 4 17 10 -7 11 8 0 7 6 3**

The positive odd number is: **3**

```
int x, count = 0;
```

```
cout << "Enter integers (0 to end):" << endl;
```

```
cin >> x ;
```

```
while ( x != 0 )
{
```

```
    if ( x % 2 == 1 )
```

```
        count ++ ;
```

```
    cin >> x ;
```

```
}
```

```
if ( count == 0 )
```

```
    cout << "No positive odd numbers!" << endl;
```

```
else
```

```
    cout << "The positive odd number is: " << count << endl;
```

```
return 0;
```

```
}
```

جملة while نوعان:  
- نوع مرتبط بعدد.  
- نوع مرتبط بتحقق شرط.

ما كان للذئب أن يكون ذئباً  
لو لم تكن الخراف خرافاً.

while statement (average of positives)

Write a program that reads positive integers and prints their total and average. The input is terminated when a negative integer entered.

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

Enter integers (-ve to end): **0 5 0 10 7 -3**  
 Total = **22**  
 Average = **7.333333**

```
int main ()
```

```
{
```

```
    int x, sum = 0, count = 0;
```

```
    double avg;
```

```
    cout << "Enter integers (-ve to end): ";
```

```
    cin >> x ;
```

```
    while ( x >= 0 )
```

```
    {
```

```
        if ( x != 0 )
```

```
        {
```

```
            sum += x ;
```

```
            count ++ ;
```

```
        }
```

```
        cin >> x ;
```

```
    }
```

```
    if ( count == 0 )
```

```
        avg = 0 ;
```

```
    else
```

```
        avg = sum * 1.0 / count ;
```

```
    cout << "Total = " << sum << endl;
```

```
    cout << "Average = " << avg << endl;
```

```
    return 0;
```

```
}
```

x	total = 0	count = 0
0	0	0
5	5	1
0	5	1
10	15	2
7	22	3
-3		

تستمر while في التنفيذ حتى يقوم المستخدم بإدخال صفر.

قبل التقسيم على قيمة يجب التأكد أنها ليست صفرية.

إذا ابتسم المهزوم فقد  
المنتصر لذة الفوز.

Event-Control while (*maximum integer*)

Write a program that keeps reading integers until 1 is entered and prints their maximum.

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

```
int main ( )
{
    int x, max ;
```

```
    cout << "Enter integers (-1 to end):" << endl;
```

```
    cin >> x;
```

```
    max = x;
```

```
    while ( x != -1 )
```

```
    {
        if ( x > max )
            max = x ;

        cin >> x;
    }
```

```
    if ( max == -1 )
```

```
        cout << "No integers entered!" << endl;
```

```
    else
```

```
        cout << "Maximum: " << max << endl;
```

```
    return 0;
```

```
}
```

Enter integers (-1 to end):

6 -3 17 9 0 -1

Maximum: 17

Enter integers (-1 to end):

-1 6 -3 17

No integers entered!

x	max
6	6
-3	6
17	17
9	17
0	17
-1	

في الحالتين تنتهي loop عندما  
تصبح  $x = -1$ ، أما  $\max$  فيكون  
(-1) فقط إذا كانت أول  $x$  تساوي (-1).

x	max
-1	-1

إذا أردت أن تنتقم ممن أساء إليك بطريقة  
تجعله يتضاعل أمام نفسه، اعف عنه.

Event-Control while (*sum of digits*)

Write a program that reads an integer and prints the sum of its digits.

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

```
int main ( )
{
    int x, d, sum = 0;

    cout << "Enter an integer: ";

    cin >> x;

    while ( x != 0 )
    {
        d = x % 10;

        sum += d;

        x /= 10;
    }

    cout << "Sum: " << sum << endl;

    return 0;
}
```

Enter an integer: **4673**

Sum: **20**

Enter an integer: **516**

Sum: **12**

فكرة هذا السؤال تتكرر بعدة أشكال.

$x = x / 10$	$d = x \% 10$	$sum += d$
4673	3	3
467	7	10
46	6	16
4	4	20
0		

في حال رقم واحد يتم قراءته  
قبل بدء الـ loop ولا يتم  
وضع cin داخل الـ loop.

لا تمنع أناسا من عبور النهر  
عندما يكون في قاربك مكان.