

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

C++ Programming

Week 5

(الشرح والأفكار الرئيسية)

- *Random Functions.*
- *Void Functions.*
- *Call by value & call by reference.*
- *Local, global, static & scope variables.*
- *Recursive Functions.*

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
للتأكد من شرائك الإصدار الأحدث.

نوت السي ++

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

هناك دائما أماكن شاغرة
على القمة.

لاستلام نسخ الكترونية من نوات الموقع مجاناً (شرح وتمارين محلولة) أو (تمارين وأمثلة إضافية) أو (امتحانات سابقة) على إيميلك قم بزيارة eng-hs.net

لطلب نوات الموقع مطبوعة ملونة مجاناً من تصوير الفرع أمام هندسة أسفل صالون رنيم أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو قم بزيارة eng-hs.net

النوتات متوفرة مجاناً بالموقعين eng-hs.net, eng-hs.com

(1)

م. حمادة شعبان 260 4444 9 info@eng-hs.com

(Random Function)Find the exact output: $x = \text{rand} () ;$

أمر () rand يعطي رقم عشوائي بين (0)، (32767)

0 ... 32767

 $x = \text{rand} () \% 11 ;$ rand () % n
ينتج رقماً بين (0)، (n-1)

0 ... 10

 $x = 1 + \text{rand} () \% 100 ;$

1 ... 100

 $x = \text{rand} () \% 50 - 50 ;$

-50 ... -1

 $x = \text{rand} () \% 30 - 40 ;$

-40 ... -11

 $x = \text{rand} () \% 6 + 1 ;$

محاكاة إلقاء نرد

1 ... 6

Write down a random statement that simulates the sum of rolling of 2 dices:

إذا الناتج يتعلق بنردين فلا بد أن نستخدم () rand مرتين في المعادلة.

- 1) $\text{sum} = 2 * (1 + \text{rand} () \% 6) ;$ (✗)
- 2) $\text{sum} = 2 + 2 * \text{rand} () \% 6 ;$ (✗)
- 3) $\text{sum} = 2 + \text{rand} () \% 11 ;$ (✗)
- 4) $\text{sum} = 2 + \text{rand} () \% 6 + \text{rand} () \% 6 ;$ (✓)
- 5) All are correct. (✗)

يسهل علينا إدراك الفرص عندما تفلت من بين أيدينا.

(Random Function)Write the equivalent random equation:

x = 10 .. 100

x = 10 + rand () % 91;

x = min + rand () % [max - min + 1] ;

x = -20 .. 20

x = -20 + rand () % 41;

x = -10 .. 0

x = -10 + rand () % 11;

x = 100 .. 300

بعد كتابة جملة rand جربها بالعكس

x = 100 + rand () % 201;

Study the following program:

```
int main ( )
{
    int x;

    // srand ( 70 );
    // srand (time (NULL));

    for (int i = 1; i <= 10; i++)
    {
        x = 1 + rand ( ) % 6;

        cout << x << endl;
    }

    return 0;
}
```

في حال عدم كتابة هذا الأمر تكون
العشرة أرقام الناتجة نفسها كل مرة

seed = 1

seed = 70

seed = time

6
6
5
5
6
5
1
1
5
34
1
3
6
2
3
2
3
2
6

changes according to
current time

لا أحد يخطط للفشل، يأتي
الفشل عندما لا نخطط.

(Random Function)

Write a program that rolls two dies (n times) and prints how many sevens encountered as the sum of the two dies.

```
# include <iostream>
# include <cstdlib> // to use srand ( )
# include <ctime>   // to use time ( )
using namespace std;

int main ( )
{
    int i, r, count = 0, n;

    srand (time (0));

    cout << "Enter number of rolls: ";

    cin >> n;

    for (i = 1; i <= n; i++)
    {
        r = 2 + rand ( ) % 6 + rand ( ) % 6;

        if (r == 7)
            count ++;
    }

    cout << "Number of sevens = " << count << endl;

    return 0;
}
```

Enter number of rolls: 1000
Number of sevens = 269

في بعض إصدارات السي لا داعي لكتابتها

تذكر أن
أي count مطلوب حسابه
يبدأ من الصفر بخلاف عداد
ال loop فإنه يبدأ من (1).

الكثير ممن فشلوا لم يدركوا مدى
قربهم من النجاح عندما استسلموا.

Void Functions (Call by value & call by reference)

Study the difference between the following two programs:

int main ()

```
{
    int x = 5;
    cout << x << endl;
    change (x);
    x += 2;
    cout << x << endl;
    change (x);
    cout << x << endl;
    return 0;
}
```

x	
5	
7	

5
8
7
10
7

تغيير القيمة بالدالة لا يؤثر على قيمتها بالـ main.

void change (int x)

```
{
    x += 3;
    cout << x << endl;
}
```

x	
5	
8	
7	
10	

الفرق بين البرنامجين هو وجود علامة & (call by reference).

int main ()

```
{
    int x = 5;
    cout << x << endl;
    change (x);
    x += 2;
    cout << x << endl;
    change (x);
    cout << x << endl;
    return 0;
}
```

x	
5	
8	
10	
13	

وجود & يسبب تحديث قيمة x بالـ main عند تغييرها بالدالة.

void change (int &x)

```
{
    x += 3;
    cout << x << endl;
}
```

x	
5	
8	
10	
13	

النجاح ملك لمن يدفع الثمن.

Void Functions (Call by value & call by reference)

Find the exact output of the following:

```
int main ( )
```

```
{
```

```
    int a = 5, b = 7;
```

```
    fun (a, b);
```

```
    cout << "Total = " << b + a << endl;
```

```
    return 0;
```

```
}
```

Total = 14

Total = 10

a	b
5	7
	5

وجود (&) تسبب تعديل القيم بالـ main

```
void fun (int a, int &b)
```

```
{
```

```
    int c = a;
```

```
    a = b;
```

```
    cout << "Total = " << a + b << endl;
```

```
    b = c;
```

```
}
```

a	b	c
5	7	5
7	5	

إن أي شكل من أشكال التخطيط أفضل
من عدم التخطيط على الإطلاق.

Void Functions - Call by Reference (*Power*)

Write a program that reads two integers and call a void function **power** that computes the power of the first integer raised to the second one. The main function will print the result.

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

Enter two integers: **3 4**
The power is: **81**

```
void power (int, int, int &);
```

```
int main ( )
```

```
{
    int x, y, p;

    cout << "Enter two integers: ";

    cin >> x >> y;

    power (x, y, p);

    cout << "The power is: " << p << endl;

    return 0;
}
```

في حال إعادة قيمة واحدة من
دالة يمكن تحقيق ذلك باستخدام
call by value
.call by reference

```
void power (int x, int y, int &p)
```

```
{
    int i;
    p = 1;
    for (i = 1; i <= y; i++)
        p *= x;
}
```

لا تقوم بتعريف p مرة أخرى لأنه تم
تعريفها بالـ (header).

خطأ نكتب int مع p لأنه تم تعريفها

ماكو return لقيمة بالـ void.

أرني شخصاً بلا أهداف كي أريك أحد
أنواع الموتى الذين تتحرك أجسادهم.

Void Functions - Call by Reference (*Total-Average*)

Write a program that reads three integers and call a function (calc) that computes their total and average.

The main function will print the results

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

Enter three integers: **3 7 9**
Total = **19**
Average = **6.33333**

```
void calc (int, int, int, int &, double &);
```

```
int main ( )
```

```
{
```

```
    int x, y, z, t;
```

```
    double v;
```

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

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

```
    calc (x, y, z, t, v);
```

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

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

```
    return 0;
```

```
}
```

```
void calc (int x, int y, int z, int &s, double &v)
```

```
{
```

```
    s = x + y + z;
```

```
    v = s / 3.0;
```

```
}
```

في حال الحاجة لإعادة أكثر من
قيمة من دالة يمكن استخدام
.call by reference

كما تنجذب المياه لمجراها الطبيعي،
ينجذب النجاح للمستعدين له.

Functions (*local-global-static-scope*)

Find the exact output of the following:

```
int x = 10;
int main ()
{
    int x = 4;
    cout << x << endl;
    {
        int x = 7;
        cout << x << endl;
        x *= x;
    }
    cout << x << endl;
    fun1 ();
    fun2 ();
    fun3 ();
    fun1 ();
    fun2 ();
    fun3 ();
    cout << x << endl;
    return 0;
}
```

من أي مكان يمكن الوصول
إلى الـ global وتعديله.

x	(Global)	
10		
11		4
12		7
		4
		30
		23
		11
		29
		23
		12
		4

في حال استدعاء دالة أكثر من مرة
فإن المتغيرات من نوع static تحتفظ
بآخر قيم لها من الاستدعاء السابق.

```
void fun1 ()
{
    static int x = 30;
    cout << x-- << endl;
}
```

```
void fun2 ()
{
    int x = 23;
    cout << x++ << endl;
}
```

```
void fun3 ()
{
    cout << ++x << endl;
}
```

x	Local (static)
30	
29	
28	

x	Local (non-static)
23	
24	
23	(في كل مرة نستدعي fun2 نترد x إلى 23)
24	

يقضي معظم الناس في الإعداد
لحفلات ميلادهم أكثر مما
يقضونه في التخطيط لحياتهم.

Functions (*local-lobal-static-scope output*)

Find the exact output of the following:

```
int x = 3;
```

```
int main ( )
```

```
{
```

```
    int y = 1;
```

```
    while (x < 7)
```

```
    {
```

```
        fun (x, y++);
```

```
        cout << "total = " << x + y << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

```
void fun (int &a, int b)
```

```
{
```

```
    static int i = 0 ;
```

```
    i += 1;
```

```
    for (int k = 0; k < i; k++)
```

```
        a += fun_fun (b);
```

```
}
```

```
int fun_fun (int y)
```

```
{
```

```
    return x - y;
```

```
}
```

x	
3	
5	
8	
14	

جرب نفسك

هذا الـ output ميدتيرم سابق.

y	
1	
2	
3	

total = 7

total = 17

a	b	i	k
3	1	0	
5		1	0
5	2	2	0
8			1
14			2

y	
1	
2	
2	

تسلط البعض لا يمكن حدوثه
إلا عن طريق جبن آخرين.

Functions (*local-global-static-scope output*)

Find the exact output of the following:

```
int num, sum = 10;
```

```
int main ( )
```

```
{
```

```
    cout << "Enter 10 values:\n";
```

```
{
```

```
    int sum = 0;
```

```
    fun (sum);
```

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

```
}
```

↓ **يستخدم global**

```
fun (sum);
```

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

```
return 0;
```

```
}
```

num	sum
5	10
10	
3	
12	
3	
3	13

اختبر
مستواك

سؤال ميدتيرم سابق.

sum
0
5
15
18
30
33

Enter 10 values:

5 10 3 12 3 3 4 5 8 3

Sum = 33

Sum = 13

```
void fun (int &a)
```

```
{
```

```
    static int count = 0 ;
```

```
    while (count < 5)
```

```
{
```

```
        ++count;
```

```
        cin >> num;
```

```
        a += num;
```

```
}
```

```
count --;
```

```
}
```

a	count
0	0
5	1
15	2
18	3
30	4
33	5
10	4
13	5
	4

يعتبر مثل هذا السؤال من الأسئلة
السهلة الممتعة بالامتحانات.

اغفر لنفسك أخطائك السابقة،
لا تجعلها تقتلك.

Recursion (Factorial)

Write a program that reads a positive integer and calls a recursive function that calculates its factorial. The main function will print the result.

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

```
int rfact (int);
```

```
int main ( )
```

```
{
```

```
    int x;
```

```
    cout << "Enter a positive integer: ";
```

```
    cin >> x;
```

```
    if (x < 0)
```

```
        cout << "Invalid number!\n";
```

```
    else
```

```
        cout << "Factorial is: " << rfact (x) << endl;
```

```
    return 0;
```

```
}
```

```
int rfact (int x)
```

```
{
```

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

```
        return 1;
```

```
    return x * rfact (x - 1);
```

```
}
```

Enter a positive integer: **4**

Factorial is: **24**

Enter a positive integer: **-3**

Invalid number!

من أقوى أدوات المبرمجين.

Given

0! = 1

n! = n . (n - 1)!

Recursion

(1) ماكو loops (2) الدالة تستدعي نفسها.
(3) لابد أن تبدأ بجمللة if. (stopping step)

rfact (4)

4 * rfact (3)

3 * rfact (2)

2 * rfact (1)

1 * rfact (0)

1

الأمر ليس سيئاً كما تعتقد، وفي الصباح سيبدو الأمر أفضل.

Recursion (*Power*)

Write a program that reads two positive integers and calls a recursive function that returns the power of the first integer raised to the second.

Note: you are not allowed to use cmath library. مرفوعاً

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

```
int rpower (int x, int y)
{
    if (y == 0)
        return 1;
    return x * rpower (x, y - 1);
}
```

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

    cout << "Enter two positive integers: ";

    cin >> x >> y;

    if (y < 0)
        cout << "Invalid power!" << endl;
    else
        cout << "The power is: " << rpower (x, y) << endl;

    return 0;
}
```

Enter two positive integers: 3 4
The power is: 81

Enter two positive integers: 4 -2
Invalid power!

Given
 $x^0 = 1$
 $x^y = x \cdot x^{y-1}$

إذا تم كتابة الدالة قبل main
فلا داعي لكتابة prototype.

عمل tracing للدالة يساعد على
فهم طريقة عملها.

```

rpower (3, 4)
3 * rpower (3, 3)
3 * rpower (3, 2)
3 * rpower (3, 1)
3 * rpower (3, 0)
1

```

الجزع عند المصيبة،
مصيبة أخرى.

Recursion (*Fibonacci*)

Write a program that reads a positive integer and calls a recursive function that calculates its Fibonacci value.
The main function will print the result.

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

Enter a positive integer: 7
Fibonacci is: 13

```
int fibon (int);
```

```
int main ()
{
```

```
    int x;
```

```
    cout << "Enter a positive integer: ";
```

```
    cin >> x;
```

```
    cout << "Fibonacci is: " << fibon (x) << endl;
```

```
    return 0;
```

```
}
```

```
int fibon (int x)
```

```
{
```

```
    if (x == 0 || x == 1)
```

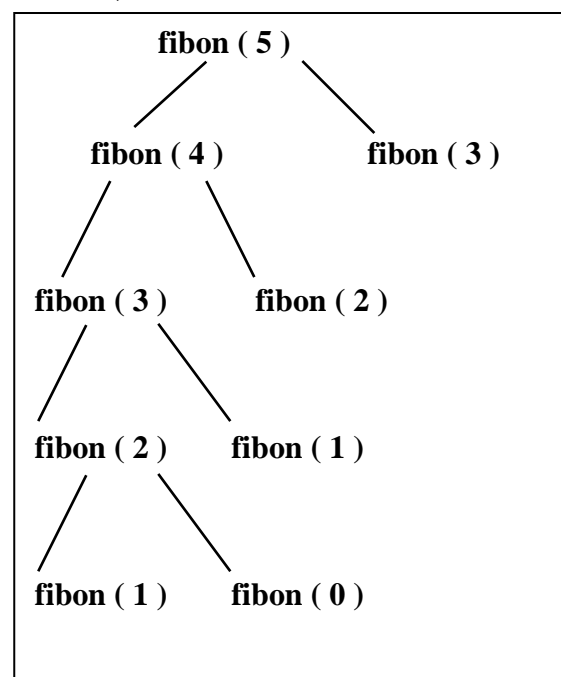
```
        return x;
```

```
    return fibon (x - 1) + fibon (x - 2);
```

```
}
```

أرقامنا	0	1	2	3	4	5	6	7	8	-----
Fibonacci	0	1	1	2	3	5	8	13	21	-----

كل رقم جديد هو مجموع الرقمين السابقين.



العجيب أن أجمل ما يحدث لنا
لا نعثر عليه، بل نتعثر به.

Recursion (*Print-Digits-Reverse*)

Write a recursive function that gets a positive integer and prints each of its digits on a separate line.

If the integer is: 1236 the function will print:

6
3
2
1

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

```
void mirror (int);
```

```
int main ( )
```

```
{
```

```
    int n;
```

```
    cout << "Enter a positive integer: " ;
```

```
    cin >> n;
```

```
    mirror (n);
```

```
    return 0;
```

```
}
```

```
void mirror (int n)
```

```
{
```

```
    if (n != 0)
```

```
    {
```

```
        cout << n % 10 << endl;
```

```
        mirror (n / 10);
```

```
    }
```

```
}
```

تذكر أن:
 $x \% 10$ = خانة الأحاد
 $x / 10$ = باقي الخانات

تكرار استخدام علامة التقسيم (/)
 يصل بأي رقم صحيح إلى الصفر.

**الغضب هو ذلك السم الذي يبتلعه
المرء آملاً أن يموت غيره.**

Recursion (*max-digit*)

Write a program that reads a positive integer and calls a recursive function that returns the maximum digit that the integer contains.
The main function will print the result.

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

Enter a positive integer: **5837**
Maximum is: **8**

```
int maxDigit (int);
```

برنامج صعب شوية.

```
int main ( )
```

```
{
```

```
    int n;
```

```
    cout << "Enter a positive integer: ";
```

```
    cin >> n;
```

```
    cout << "Maximum is: " << maxDigit (n) << endl;
```

```
    return 0;
```

```
}
```

```
int maxDigit (int n)
```

```
{
```

```
    if (n == 0)
```

```
        return 0;
```

```
    if (n % 10 > maxDigit (n / 10))
```

```
        return n % 10 ;
```

```
    return maxDigit (n / 10);
```

```
}
```

إما أن يكون أكبر رقم موجوداً
في خانة الآحاد أو أن يكون
موجوداً بأحد باقي الخانات.

حاول تسوي tracing
فإنه يسهل فهم الدالة.

الهم أحد أنواع التعذيب الذاتي للنفس،
بعض الناس يتقنون هذا الفن!

Exercise: Write a program that simulates coin tossing. Let the program toss the coin 100 times and count the number of times each side of the coin appears. Print the results. The program should call a separate function *flip* that takes no arguments and returns false for tails and true for heads.

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

```
bool flip ();
```

```
int main ()
```

```
{
    bool toss;
    int heads, tails;
    heads = 0, tails = 0;
    srand ( time ( 0 ) );
    for ( int i = 1; i <= 100; i++ )
    {
        toss = flip ();
        if ( toss == true )
            heads++;
        else
            tails++;
    }

    cout << "Heads: " << heads << ", tails: " << tails << endl;

    return 0;
}
```

```
bool flip ()
```

```
{
    if ( ( rand () % 2 ) == 1 )
        return true;
    return false;
}
```

كل run تعطي
قيمة مختلفة

Heads: 55, tails: 45

Heads: 47, tails: 53

وقعاتنا تمثل نبوءات مستقبلنا.

Exercise: Write a program that helps an elementary school student learn multiplication. Use rand to produce two positive one-digit integers. It should then type a question such as

How much is 6 times 7?

The student then types the answer. Your program checks the student's answer. If it is correct, print "very good!". If the answer is wrong, print "No. Please try again", until the student finally gets it right.

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

```
bool check ( int, int, int );
```

```
int main ( )
```

```
{
    int x, y, z;
    bool result;

    srand ( time ( 0 ) );
    x = 1 + rand ( ) % 10;
    y = 1 + rand ( ) % 10;

    cout << "How much is " << x << " times " << y << "? ";
    cin >> z;
    result = check ( x, y, z );

    while ( result != true )
    {
        cout << "No, Please try again: ";
        cin >> z;
        result = check ( x, y, z );
    }
    cout << "Very good!\n";

    return 0;
}
```

برنامج تعليم ضرب رقمين إذا
سويته وشغلته ستشعر بمتعة
(أكو طلبه بطلوا يشعرون)

إذا جاءك الطعن من الخلف
فهذا يعني أنك في المقدمة.

```
bool check (int x, int y, int z)
```

```
{
    if ( z == x * y )
        return true;
    return false;
}
```

```
// we can replace the 3 lines with: return ( z == x * y );
```

How much is 9 times 10? 90

Very good!

How much is 5 times 5? 10

No. Please try again: 20

No. Please try again: 25

Very good!

بعض طلاب هندسة يحتاجون يتذكرون
جدول الضرب وإذا تبي تتأكد: اسأل
صديقك ($9 * 7 = ?$)

إذا كان السفهاء والحمقى
يقررون كيف يعيشون فمن
المؤكد أن هذا الحق مكفول لنا.

Exercise: Write a recursive function gcd that returns the greatest common divisor of x and y.

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

int gcd ( int, int );

int main ( )
{
    int x, y;

    cout << "Enter two integers: ";
    cin >> x >> y;

    cout << "Greatest Common Divisor is: " << gcd ( x, y ) << endl;

    return 0;
}

int gcd ( int x, int y )
{
    if ( y == 0 )
        return x;

    return gcd ( y, x % y );
}
```

Enter two integers: **36 12**
Greatest Common Divisor is: **12**

Enter two integers: **15 4**
Greatest Common Divisor is: **1**

Enter two integers: **18 24**
Greatest Common Divisor is: **6**

إذا كنت تحب شيئاً أطلقه، فإن
عاد إليك فهو ملكك للأبد.