

السنة : الثانية

الفصل : الأول

التاريخ : 2013/10/9

كلية العلوم قسم الرياضيات - جامعة دمشق

المقرر : برمجة وخوارزميات

المحاضرة : (3)

مثال :

اكتب برنامج بلغة C++ يقوم بطباعة عبارة ما.

```
#include <iostream.h >

int main () {

cout <<"this is my first program in c++";

return 0;

}
```

مثال :

اكتب برنامج يقوم بإدخال عددين صحيحين ويطبع ناتج جمعهما

```
#include <iostream.h>

int main () {

int a,b;

cin>>a>>b;

cout<<a+b;

return 0;

}
```

مثال :

اكتب برنامج يقوم بإدخال عددين صحيحين ويطبع ناتج ضربهما  
نفس البرنامج السابق ولكن نضع عوضاً عن عملية الجمع عملية الضرب

ملاحظة :

يمكننا في البرامج السابقة إدخال متحول جديد وإسناد عملية الجمع/الضرب له وطباعته.

ملاحظات عامة :

- 1- كل تعليمات لغة ++C تكتب بأحرف صغيرة.
- 2- أي تعليمة من تعليمات اللغة تكتب دائماً ضمن دوال ولا يمكن كتابة تعليمات خارج الدوال.

<<ملاحظات هامة>>

(1) معامل الزيادة (++):  
بشكل عام..

$$a ++ \Leftrightarrow a = a + 1$$

$$++ a \Leftrightarrow a = a + 1 \quad \text{أو}$$

أما عند المعالجة فثمة اختلاف شاسع بينهما

$$\text{لنفرض أن } a = 15, b = 9$$

ولنحسب :

$$C = ++ a - b = 16 - 9 = 7, a = 16$$

$$D = a ++ - b = 15 - 9 = 6, a = 16$$

ملاحظات :

- عندما يأتي معامل الزيادة قبل المتحول نقوم بتعديل نتيجة المتحول ثم ننجز العملية الحسابية.
- عندما يأتي معامل الزيادة بعد المتحول نقوم بإنجاز العملية على القيم الراهنة للمتحويلات ثم نعدل قيمة المتحول.

(2) معامل النقصان (- -) :

بشكل عام..

$$- - a \Leftrightarrow a = a - 1$$

$$a - - \Leftrightarrow a = a - 1$$

أو

أما عند المعالجة فهناك فرق بينهما

$$a = 15, b = 9 \text{ لنفرض}$$

ولنحسب

$$C = - - a - b = 14 - 9 = 5, a = 14$$

$$D = a - - - b = 15 - 9 = 6, a = 14$$

إذاً الحال هنا كما في معامل الزيادة ..

- عندما يأتي معامل النقصان قبل المتحول نعدل قيمته ثم ننجز العملية

- وعندما يأتي معامل النقصان بعد المتحول ننجز العملية ثم نعدل قيمة المتحول

بعض الرموز في اللغة :

$$a += 1 \Leftrightarrow a = a + 1$$

$$a -= 3 \Leftrightarrow a = a - 3$$

$$x *= y \Leftrightarrow x = x * y$$

$$a /= 5 \Leftrightarrow a = a / 5$$

<< تعليمات لغة C++ >>

1) التعليلة الشرطية البسيطة (if) :

شكلها العام

$\text{if}(\underbrace{\text{condition}}_{\text{الشرط}})\{\underbrace{\text{statements}}_{\text{التعليمات}};\}$

وهي تعني أنه إذا تحقق الشرط الموجود بعد كلمة if وبين قوسين ( ) عندئذٍ نفذ التعليمات {statements;}

ملاحظات :

1) عند وجود تعليلة واحدة داخل قوس {statements;} يمكن الإستغناء عن القوسين { } وإذا وضعناهما فلا مشكلة في ذلك.

2) شرط if هو عبارة عن قضية منطقية بسيطة أو مركبة.

مثال :

اكتب برنامج يقوم بإدخال عدد صحيح وطباعته إذا ما كان موجبا.

```
#include <iostream.h>

in main () {

int n;

cout<<"input the value of un integer of number = ";

cin>>n;

if(n>0) cout <<n;

return 0;

}
```

2) تعليمة (if..else) :

شكلها العام

```
if (condition) {statements1;}
```

```
else {statements2;}
```

وهي تعني أنه إذا تحقق الشر نفذ التعليمات {statements1;} وإلا إذا لم يتحقق نفذ التعليمات {statements2;}.

مثال :

اكتب برنامج يقوم بإدخال عدد صحيح وطباعة قيمته المطلقة.

```
#include <iostream.h>
```

```
int main () {
```

```
int x;
```

```
cin>>x;
```

```
if (x<0) cout <<-x;
```

```
else cout <<x;
```

```
return 0;
```

```
}
```

ملاحظة : كل تعليمة في لغة C++ تنتهي بـ (;)

(3) التعلیمه الشرطیه (if\_else\_if)

شکلها العام..

```
if (condition1) {statements1;}  
  
    else if (condition2) {statements2;}  
  
    else if (condition3) {statements3;}  
  
    :  
  
    :  
  
    else if (condition) {statementsn;}  
  
else {statementsn+1;}
```

مثال (2) :

اكتب برنامج يقوم بحساب قيمة الدالة f المعرفة كما يلي :

$$f(n) = \begin{cases} n^4 - 4 & : n < 0 \\ 1 & : n = 0 \\ n - 2 & : n > 0 \end{cases}$$

حيث n عدد صحيح.

```
#include <iostream.h>  
  
int main () {  
  
int n;  
  
cout <<"input n= ";  
  
cin >>n;  
  
if (n<0)
```

```
cout <<(n*n*n*n)-4;  
  
else if (n==0)  
  
cout <<1;  
  
else cout <<n-2;  
  
return 0;  
  
}
```

مئال (2) :

اكتب برنامج يقوم بحساب قيمة الدالة f المعرفة كما يلي :

$$f(n) = \begin{cases} n - 1 & : n > 0 \\ 0 & : n = 0 \\ n^3 - 3 & : n < 0 \end{cases}$$

على نفس مبدأ التمرين السابق.

ملاحظة :

(==) ترمز إلى عملية المقارنة

(=) ترمز إلى عملية الإسناد

### المحاورة (3)

أمور عامة تتعلق بالطباعة :

"\t"

الهدف منها ترك 8 فراغات بين تعليمة وأخرى

مثال :

a=13 , b=9

(( شاشة التنفيذ ))

`cout <<a<<b;`

ومن أجل ذلك نكتب :

`cout <<a<<"\t"<<b;`

وبشكل أوضح من ذلك نكتب :

`cout <<"a="<<a<<"\t"<<"b="<<b;`

139	
13	9
a=13	b=9

endl  $\Leftrightarrow$  "\n"

في لغة ++C تعلن عند ظهورها انتهاء السطر والانتقال لسطر جديد

في لغة C تعني انتقال لسطر جديد

### المماخضة (3)

مثلاً :

$$a=13, b=9$$

```
cout <<a<< "\n"<<b
```

```
[cout << a << endl  
 cout << b;
```

وكلتاها لهما نفس المعنى وتظهران بنفس الشكل على شاشة التنفيذ :

13

9

الحلقات التكرارية (loops)

1 حلقة (for) :

شكلها العام هو :

for (مقدار تغيير العداد ; شرط الاستمرار ; قيمة ابتدائية = اسم عداد الحلقة)

{ تعليمات ;

}

ملاحظة :

يجب أن يكون العداد من النوع المعداد ((بشكل مبدئي سنستخدم إما int أو char))

شرط الاستمرار : عبارة عن قضية منطقية يتضمن بشكل غير مباشر شرط التوقف

قف → استمر

مقدار التغيير : المقدار بين قيمة وقيمة تليها (إما تصاعدياً أو تنازلياً))

القيمة الابتدائية : وهي القيمة التي يجب أن تبدأ وهي أول قيمة

فننتقل إما منها وللأكبر

أو منها وللأصغر

### المحاضرة (3)

معنى حلقة for :

من أجل جميع قيم عداد الحلقة التي تبدأ بالقيمة الابتدائية وتحقق شرط الاستمرار , كرر تنفيذ التعليمات المحصورة بين قوسين for علماً أن عداد الحلقة يتغير وفق مقدار التغير المحدد.

مثال :

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

ملاحظة :

- في لغة C++ حيثما نحتاج متحول جديد نعرفه في المكان الذي نحتاجه فيه بعكس لغة الباسكال التي تفرض علينا تعريف كل المتحولات بعد var فقط.

... انتهت المحاضرة (3) ...