



المحاضرة الخامسة

# البرمجة 1

## ( *Programming Language 1* )

إعداد

الدكتور المهندس فراس الزين

## الكلمات المفتاحية

جمل التحكم , جمل شرطية , البرنامج , الشرط , إذا , إلا , تعبير رياضي , جملة التبديل ,  
الحالة , الحالة الافتراضية , القطع , التابع , الحلقة , شكل عام , من أجل , طالما , كرر ,  
القيمة , تزايد .

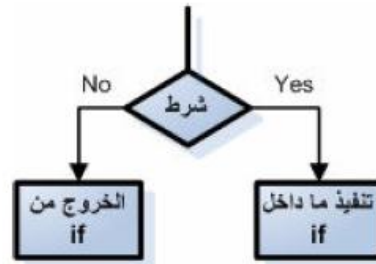
*Control statement , Condition statement , program , condition , if ,  
else , expression , switch , case , default , break , function , loop , public  
formula , for , while , do , value , increment .*

## جمل التحكم Control Statements

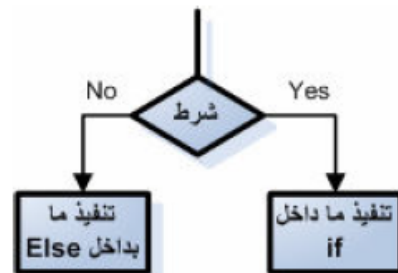
الجمل الشرطية Condition Statements:

(١) جملة IF:

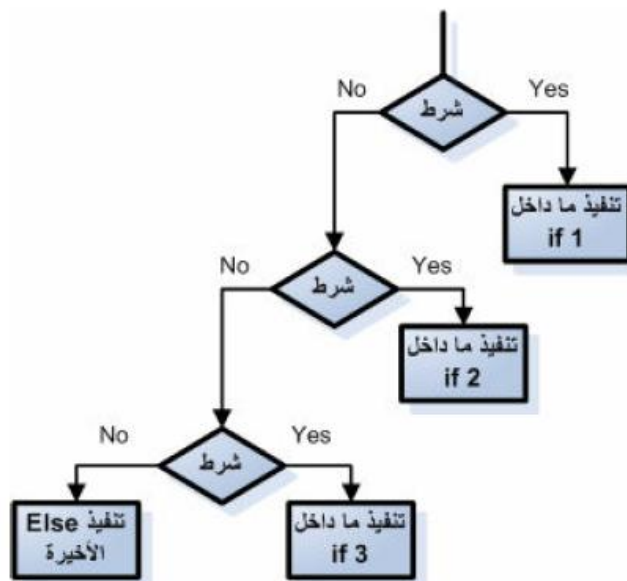
```
if ( Expression )  
{  
    Statements;  
    Statements;  
    .....;  
}
```



```
if ( Expression )  
{  
    Statements;  
    Statements;  
    .....;  
}  
else  
{  
    Statements;  
    Statements;  
    .....;  
}
```

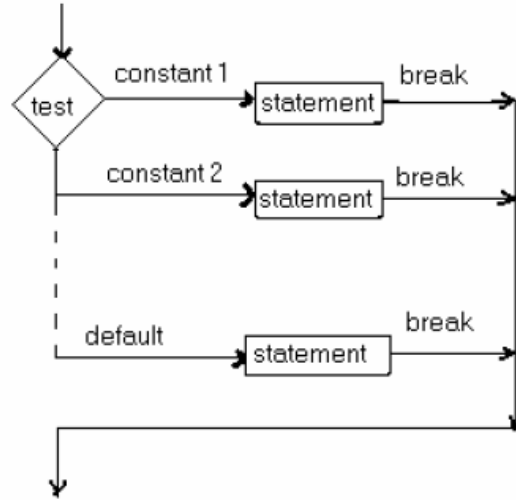


```
if ( Expression ) {  
    Statements;  
    Statements;  
    .....;  
}  
else if ( Expression ) {  
    Statements;  
    Statements;  
    .....;  
}  
else {  
    Statements;  
    Statements;  
    .....;  
}
```



## (٢) جملة switch:

```
switch ( var )  
{  
  case 1 :  
    Statements1;  
  break;  
  
  case 2:  
    Statements;  
  break;  
  
  default:  
    Statements;  
}
```



ملاحظة :

فائدة break بعد كل case أنها توقف عمل switch بعد تنفيذ جملة case المناسبة، وإذا لم يتم كتابة الكلمة break فإن البرنامج سينتقل إلى case التالية وينفذها حتى لو لم ينطبق الشرط var عليها. يمكن كتابة الجزء الأخير الخاص ب default ويمكن عدم كتابته، حيث يتم تنفيذ الجمل داخل default عندما لا تتحقق أي من الحالات "cases" السابقة، فهي تشبه else الأخيرة في جملة if.

مثال:

```
1. int r = 3  
2. switch ( var )  
3. {  
4.   case 1 : cout << "one\n";   break;  
5.   case 2: cout << "two\n";   break;  
6.   case 3: cout << "three\n"; break;  
7.   default: cout << "Error!\n";  
8. }
```

ملاحظة:

إذا كان المتغير حرفي نستخدم علامة الاقتباس المفردة مثل: 'y' case وإذا كان نصي نستخدم علامة الاقتباس المزدوجة مثل: "yes".case

تمرين :

اكتب برنامج لمعرفة نوعية العدد (سالبة أو موجب أو غير ذلك).

## دوال الدوران :Loops Functions

### :For (١)

تحتاج دالة for إلى عداد (رقم تبدأ منه الدوران ورقم تنتهي إليه) لكي تنفذ الدوران ومقدار الزيادة(١).

الشكل العام Public formula:

```
for ( initialization_value; condition; Increment or Decrement)  
Statements...
```

مثال :

النتيجة:	الкод:	1. for ( int i = 0; i <=3; i++) 2. { 3. cout << "i value is : " << i << endl; 4. }	i value is: 0 i value is: 1 i value is: 2 i value is: 3
----------	--------	---	--

### :While (٢)

تحتاج دالة while إلى شرط يحدد استمرارها أو توقفها، فهي ستستمر بلا توقف طالما الشرط متحقق.

الشكل العام Public formula:

```
while (condition)  
Statements...
```

مثال :

النتيجة:	الкод:	1. int w = 3; 2. while ( w <=3 ) 3. { 4. cout << "value is : " << w << endl; 5. w++; 6. }	value is: 0 value is: 1 value is: 2 value is: 3
----------	--------	--	--

ملاحظة: يمكن الاستغناء عن الأقواس {} الخاصة بدالة for و while و if إذا كانت الجملة التي تنفذها تتكون من سطر واحد.

### ٣) do while :

الشكل العام Public formula:

```
do
{
    Statements...
}
while (condition)
```

مثال :

<pre>1. int w = 0; 2. do 3. { 4.     cout &lt;&lt; "value is : " &lt;&lt; w &lt;&lt; endl; 5.     w++; 6. } 7. while ( w &lt;=3 );</pre>	الكود:	<pre>value is: 0 value is: 1 value is: 2 value is: 3</pre>	النتيجة:
--	--------	--	----------

لكن do while تقوم بتنفيذ الكود مرة واحدة حتى لو كان الشرط خاطئاً:

<pre>1. int w = 0; 2. do 3. { 4.     cout &lt;&lt; "value is : " &lt;&lt; w &lt;&lt; endl; 5. } 6. while ( w &gt;0 );</pre>	الكود:	<pre>value is: 0</pre>	النتيجة:
---	--------	------------------------	----------

### القيمة التزايدية Increment value (معنى ++i):

تعني زيادة المتغير  $i$  بمقدار واحد فقط وهو اختصار للجملة التالية:

$i = i + 1;$

ويمكن زيادة المتغير  $i$  بأي مقدار نريد بالشكل التالي:

$i += 5;$

وهذا يكافئ السطر التالي:

$i = i + 5;$

أو إنقاص المتغير  $i$  بأي مقدار نريد بالشكل التالي:

$i -= 5;$

وهذا يكافئ السطر التالي:

$i = i - 5;$

### الفرق بين ++i و ++i :

مثال ١:

```
cout << i++;
cout << ++i;
```

طباعة  $i$  ثم زيادته بمقدار واحد.  
زيادة  $i$  بمقدار واحد ثم طباعته.

مثال ٢:

```
int z, i;
i = 1;
z = 5 * i + i++;
```

```
int z, i;
i = 1;
z = 5 * i + ++i;
```

التنفيذ :

التنفيذ :

$z = 5 * 1 + 1;$

$z = 5 * 1 + 2;$

قيمة Z:

قيمة Z:

6

7

تمرين :

عمل برنامج يقيم بطباعة مثلثات باسكال باستخدام دالتي for فقط.

```

      1
     1 2
    1 2 3
   1 2 3 4
  1 2 3 4 5
 
```

للتسهيل يمكن عملها باستخدام رمز النجمة (\*) فقط:

```

      *
     * *
    * * *
   * * * *
  * * * * *
 
```