الجزء النظري (المحاضرة رقم 06)

الأولوية في العمليات الحسابية
(precedence of Arithmetic Operations)

ان ترتيب الاسبقية في تنفيذ العمليات الرياضية وكما يجريها مترجم (C++) بالنسبة للأدوات الحسابية هي على النحو التالي :

الأداة	رقم الأولوية
++ أو --	1
-	2
* أو / أو %	3
+ أو -	4
=	5
++ أو -- (المتأخرة بعد العدد)	6

الأولويات في التنفيذ:

1) والزيادة والنقصان عندما تاتي قبل العدد.
2) الاقواس.
3) إشارة السالب.
4) القسمة وباقي القسمة والضرب.
5) الجمع والطرح.
6) المساواة.
7) الزيادة والنقصان المتأخر بعد العدد.

ملاحظة مهمة: في حال وجود عمليتين لهما نفس الأولوية نبدأ بتنفيذ العملية الأقرب الى جهة اليسار. وعند استعمال الاقواس لاي تعبير فان الاقواس تأخذ الأولوية الأولى في التنفيذ قبل (الزيادة او النقصان).

مثـــــال: وضح تسلسل تنفيذ العمليات الحسابية في المعادلة ادناه:

(X+y/z*a)

يكون تسلسل أولويات التنفيذ حسب الخطوات التالية:

1) العملية الأولى: القسمة y/z
2) العملية الثانية: a*(y/z)
3) العملية الثالثة: جمع الناتج في خطوة 2 مع x وتكون النتيجة X+y/z

نلاحظ ان تنفيذ العمليات الحسابية بدأ من جهة اليسار الى اليمين، وتعطى الأولوية حسب الجدول أعلاه، وفي المثال تكون اعلى أولوية الى (القسمة) ثم الأقل أولوية هي عملية الضرب ثم عملية الجمع.

الأدوات العلاقية والمنطقية
(Relational and Logical Operations)

يرجع اسم الأدوات العلاقية الى العمليات المختصة بالقيم التي بينها علاقات وهو اجراء عمليات مقارنه منطقية بين كميات حسابية او رمزية، وتكون نتيجته منطقية وهي اما (نعم true) او (لا false). ويكثر استخدام التعابير المنطقية في الجمل الشرطية ونبين في ادناه جدول الأدوات العلاقية والادوات المنطقية:

الأدوات العلاقية:

الأداة	معناها
>	اكبر من
<	اصغر من
>=	اكبر من أو يساوي
<=	اصغر من أو يساوي
= =	يساوي
!=	لا يساوي

التعبير المنطقي (X==y) يكون جوابه اما نعم او لا.

تعامل النتيجة (لا false) على انها صفر (0) وتكون النتيجة (نعم true) أي قيمتها غير الصفر والمتعارف عليه ان قيمتها (1) .

الادوات المنطقية:

الأداة	معناها
&&	And (حرف العطف واو)
||	Or (حرف العطف أو)
!	Not (للنفي) أداة أحادية unary

مثـــال: نفترض ان (int a=b=3) تكون النتيجة للتعابير التالية كما يأتي:

فان التعبير a < 3 نتيجته false أي 0
التعبير a<=3 نتيجته true أي 1
التعبير a>b نتيجته false أي 0
التعبير a!=b نتيجته false أي 0
التعبير a= = b نتيجته true أي 1

ملاحظة : الأولويات تكون على نفس الترتيب (من اليسار الى اليمين).

الأدوات الدقيقة
Bowties Operators

من مميزات لغة البرمجة (C++) عن سائر اللغات الاخرى البرمجية استخدامها لأدوات دقيقيه تتعامل مع (bit) وتستعمل هذه الأدوات مع المعطيات (char) و (int) وتستخدم مع غيرها .
وهذه الأدوات هي :

الشرح	التعليمة
Not	~
إزاحة الى اليسار	>>
إزاحة الى اليمين	<<
Xor	^
And	&
Or	|

وتكون الأولوية حسب ترتيب الجدول أعلاه .
مثـــــال :

اذا كان لدينا (X) ممثل بالنظام الثنائي ، المطلوب اوجد قيمة نفي (X) أي (~X) .

قيمة (X)

0	0	0	1	1	0	1	0

يكون الجواب:

1	1	1	0	0	1	0	1

جمل التحكم والشرط والتكرار
(Program control, condition and loop statement)

1) الجمل الشرطية (Condition statement):

تقسم الجمل الشرطية الى نوعين هما:

1- جملة إذا الشرطية (if – statement).

2- جملة التوزيع (switch – case).

2) حلقات التكرار (loops statement):

وهي مجموعة من الجمل تستعمل لتكرار تنفيذ مجموعة من الأوامر لعدد محدد من المرات، وفي ادناه نوعين من حلقات التكرار هي:

1- حلقات for (for ...loops).

2- حلقات while (while ... condition)

وتستخدم أوامر للتحكم بعملية التكرار هي:

a. امر التوقف break او return

b. امر الاستمرار continue

الجمل الشرطية (Condition statement):

وهي عبارة عن جمل تتحكم في تنفيذ خطوات البرنامج من خلال شرط او مجموعة شروط ومن هذه الجمل هي :

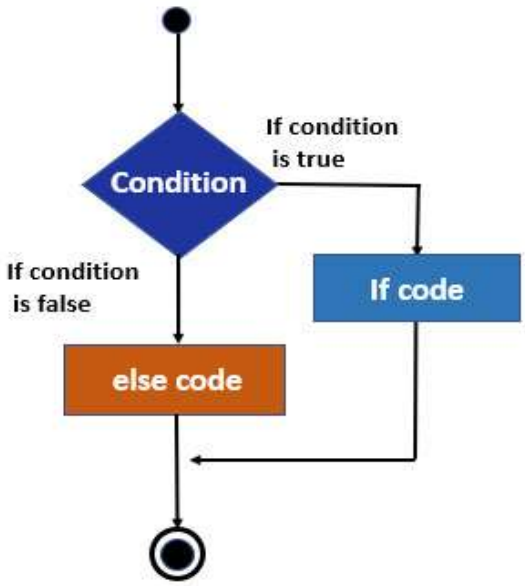
1) جملة إذا الشرطية (if) والتي تأخذ ثلاث اشكال :

a. العبارة الشرطية البسيطة (if – statement)

تستخدم العبارة البسيطة في اختبار شرط اذا تحقق الشرط يتم تنفيذ تعليمة او مجموعة من التعليمات والصيغة العامة للعبارة الشرطية البسيطة في حالة تنفيذ تعليمة واحدة :

```
1- if (condition)
      statement;

if (x==100)
      cout<<"x is 100";
```



والمثال ادناه يمثل الصيغة العامة للعبارة الشرطية البسيطة في حالة تنفيذ مجموعة من التعليمات

```
2- if(condition)
    {
      statement 1;
      statement 2;
      ....................
      statement n;
    }
```

```
if(x==100)
  {
    cout<<"x is ";
    cout<<"100";
  }
```

امثلة :

1) مثال رقم (1)

```
#include <iostream>
#include <conio.h>
using namespace std;
int main ()
{
float number;  /* هذا المتغير يمثل الرقم الذي سيدخله المستخدم */
cout << "Enter a number: ";
cin >> number;  /* انتظار المستخدم ليدخل الرقم */
if (number == 5)  /* هل number يساوي 5 */
{
cout << "The number is = 5 .";
}

getch();
return 0;
```

```
}
```

2) مثال رقم (2) :

برنامج يطلب من المستخدم ادخال رقم , فإذا كان الرقم أكبر من "5" فإن البرنامج يخبره أن الرقم أكبر من "5" , اذا كان أصغر من "5" يخبره البرنامج أن الرقم أقل من "5" , ايضاً اذا كان الرقم يساوي "5" فإن البرنامج سيخبره أن الرقم يساوي "5." "

```
#include <iostream>
#include <conio.h>
using namespace std;
int main ()
{
float number;  /* هذا المتغير يمثل الرقم الذي سيدخله المستخدم
*/
cout << "Enter a number: ";
cin >> number;  /* انتظار المستخدم ليدخل الرقم */
if (number > 5)  /* هل number أكبر من 5 */
{
cout << "The number is greater than 5 .";
}
else if (number < 5)  /* هل number أصغر من 5 */
{
cout << "The number is less than 5 .";
}
else  /* اذا لم يكن أكبر من 5 ولا اصغر منه اذن فهو يساويه */
{
cout << "The number is 5 .";
}

getch();
return 0;
}
```