الجزء النظري (المحاضرة رقم ٠٥)

القيم الابتدائية الثابتة والمتغيرة (Initialization & Const):

- **القيم الابتدائية:** وهي القيم التي تسند للمتغير بمجرد تعريفه وهي تاتي بطريقتين هما :

١) القيم المتغيرة :

وهي التي يمكن تغييرها في البرنامج من قبل المبرمج او اثناء التشغيل باستخدام الامر (cin) .

```
1. int x = 5 ;
2. x = 10 ;
3. x = 20 ;
4. cin >> x ;
```

تعريف متغير x يحمل قيمة ابتدائية 5.
إسناد قيمة جديدة لـ x.
تغيير القيمة السابقة بقيمة جديدة أخرى.
تغيير القيمة أثناء التشغيل من قبل المستخدم.

٢) القيم الثابتة (const):

لايمكن تغييرها باي حال من الأحوال والفائدة منها لحماية القيم التي نحتاجها كما هي ولا نريد ان يتم تغيير قيمتها بالخطأ مثل قيمة النسبة الثابتة (3.14).

```
1. const int x = 5;
2. x = 10 ;    // Error
```

تعريف ثابت يحمل قيمة ثابتة 5.
إسناد قيمة جديدة يولد خطأ في تنفيذ البرنامج.

التعليقات Comments:

وهي عبارة عن توضيحات يكتبها المبرمج لاتدخل في تركيب البرنامج أي لاينفذها المترجم . وتاتي على شكلين هما :

١) تعليق لسطر واحد:

```
1. // This is a comment
2. // And this is another comment
```

٢) تعليق الاسطر المتعددة:

```
1. /*
2. This is a comment
3. In tow lines
4. */
```

```
1. /* This is
2. a comment
3. in three lines */
```

أنواع المكتبات (Libraries type):

المكتبة (IOSTREAM):

تم دمج مكتبتين ضمن هذه المكتبة هما :

- Istream مكتبة الادخال .
- Ostream مكتبة الإخراج .

أوامر الادخال والإخراج:

تحتوي مكتبات الادخال والإخراج على دوال منها (cin) و (cout) ويتم استخدامها كما يلي :

```
1. cin >> x ;
2. cout << x ;
3. cout << " نص " ;
4. cout << ' c ' ;
```

إدخال قيمة للمتغير x من قبل المستخدم:

إخراج قيمة المتغير إلى الشاشة:

أرقام ورموز وحروف (انجليزية) باستخدام شرطة مزدوجة " "

حرف واحد باستخدام شرطة مفردة ' '

مثال ١:

```
1. #include <iostream.h>
2. Main()
3. {
4.     int x ;
5.     cin >> x ;
6.     cout << " X value is: " << x ;
7. }
```

مثال ٢:

```
1. #include <iostream.h>
2. Main()
3. {
4.     int x , y ;
5.     cin >> x >> y ;
6.     cout << " first value is: " << x  << " second value is: " << y ;
7. }
```

الثوابت الرمزية:

هي رموز محجوزة تعبر عن الحروف غير المطبوعة وتستخدم مع الدوال مثل (cout) وتكون محصورة بين علامات الاقتباس (المفردة او المزدوجة) وكما في المثال ادناه :

المحرف	المعنى	توضيح
\n	New line	سطر جديد
\t	8 Spaces (Tap)	٨ مسافات فارغة
\b	Backspace	الرجوع للخلف
\a	Sound "beep"	إصدار صوت من الجهاز

مثال:

```
1 cout << '\n';
2 cout << "Ahmed \t 20";
3 cout << "khaled\nSaleh";
```

الناتج: النزول إلى سطر جديد فارغ

الناتج :

Ahmed 20

الناتج:

khaled
Saleh

دوال تقوم بعمل الثوابت الرمزية تستخدم مع دالة (cout) :

الدالة	المعنى	توضيح
endl	New line	سطر جديد
ends	8 Spaces (Tap)	٨ مسافات فارغة

مثال:

```
1  cout << "Ahmed" << ends << "20";
2  cout << "khaled" << endl << "Saleh";
```

الناتج :

```
Ahmed    20
```

الناتج:

```
khaled
Saleh
```

الأدوات المستخدمة في لغة (C++) :

يوجد في لغة (C++) ثلاث أنواع من الأدوات هي :

١) الأدوات الحسابية
٢) الأدوات المنطقية او العلائقية
٣) الأدوات الدقيقة

١) الأدوات الحسابية (Arithmetic Operators):

تسمح لغة (C++) باستعمال الأدوات الحسابية مثل الجمع والطرح والضرب والقسمة كاللغات الأخرى . الا ان عملية الرفع الى الاس ليس لها أدوات مباشرة .

كما وتختلف عملية القسمة (/) في لغة (C++) حيث انها تقوم بقسمة الاعداد وان أي جزء كسري ينتج عن القسمة يهمل مهما كان كبير والأدوات الحسابية مبينه في ادناه :

الأداة	وظيفتها
-	للطرح أو كأشاره سالبة
+	للجمع
*	للضرب
/	للقسمة
%	لباقي القسمة الصحيحة
- -	للنقصان
+ +	للزيادة

ملاحظة :

عند استخدام العمليات الرياضية يجب الحذر من ان تتعدى قيمة النتيجة من أي عملية حسابية الحدود المرسومة لنوع المتغير الناتج . وذلك لان كل نوع من أنواع المتغيرات يمتلك حدود لايجوز تعديها وعند تجاوزها يكون هناك خطأ بالنتائج . كذلك يجب ان نعرف اذا تعدت النتائج الحدود المسموح بها فهذا يسمى خطأ من نوع (overflow) وفي هذا النوع من الأخطاء المترجم لايعطي أي رسالة خطأ تعلم المبرمج بالخطأ الحاصل.

```
1.  int x = 3 ;
2.  int z = 7 ;
3.
4.  cout << "z + x= " ;
5.  cout << z + x ;
```

الناتج هو "z + x= "
الناتج هو 10

الزيادة والنقصان (Increment and Decrement):

من المميزات التي تمتاز بها لغة (C++) هي استعمالها للأداتين (++) و (--) لزيادة القيم بمقدار واحد او انقاصها بمقدار واحد وكما مبين في ادناه :

```
a++;
++a;
```

مثال :

```
#include <iostream>
using namespace std;
main()
{
    int k,y,z;
    k=y=z=0;
cout << "****step 01 *******"<< endl;
cout << " value of k="<<k << endl;
cout << " value of y="<<y << endl;
cout << " value of z="<<z << endl;
cout << "********************"<< endl;
k=++y + ++z;
cout << "****step 02 *******"<< endl;
cout << " value of k="<<k << endl;
cout << " value of y="<<y << endl;
cout << " value of z="<<z << endl;
cout << "********************"<< endl;
k=y++ + z++;
cout << "****step 03 *******"<< endl;
cout << " value of k="<<k << endl;
cout << " value of y="<<y << endl;
cout << " value of z="<<z << endl;
cout << "********************"<< endl;
k=++y + z++;
cout << "****step 04 *******"<< endl;
cout << " value of k="<<k << endl;
cout << " value of y="<<y << endl;
cout << " value of z="<<z << endl;
cout << "********************"<< endl;
k=y-- +--z;
cout << "****step 05 *******"<< endl;
cout << " value of k="<<k << endl;
cout << " value of y="<<y << endl;
cout << " value of z="<<z << endl;
cout << "********************"<< endl;
}
```

```
****step 01 *******
 value of k=0
 value of y=0
 value of z=0
********************
****step 02 *******
 value of k=2
 value of y=1
 value of z=1
********************
****step 03 *******
 value of k=2
 value of y=2
 value of z=2
********************
****step 04 *******
 value of k=5
 value of y=3
 value of z=3
********************
****step 05 *******
 value of k=5
 value of y=2
 value of z=2
********************
```