ب- أنواع البيانات

أى برنامج مهما كان لابد وأن يتعامل مع البيانات ، وتختلف نوعية البيانات طبقاً للغرض فعلى سبيل المثال يتم استخدام أعداد صحيحه للوصول لبيانات مخزنة مصفوفه فالبيانات هنا التى معنا والتى تتعامل مع الارقام الصحيحه والارقام بالكسور العشريه

فالبيانات ذات الخصائص المختلفه يتم معالجتها كلّ بطريقه مختلفه ، فالارقام الصحيحه يتم معالجتها بسرعه فى حين معالجه الارقام العشريه يحتاج الى عمليه اضافيه فرعيه وبرغم تداخل تلك الارقام مع البيانات العائمه (بالفاصلة العشرية) فيتم معالجتهم بطريقة أبطئ من الارقام الصحيحه

البيانات المتسلسله تستغرق الوقت الاطول بسب ديناميكيه ذاكرة الكمبيوتر فى تحديد المواقع

انواع البيانات الرئيسية التى نتعامل معها هى :

* الأعداد
* القيم المنطقيه

* الحروف
* المتسلسلات
* الارقام بالفاصلة العشرية (ذات العلامة العشرية)
* الالوان
* الوقت والتاريخ

لاحظ ان الوقت والتاريخ والالوان تستعمل فقط لتسهيل وتوضيح البيانات للمستخدم عند تشغيله للاكسبيرت أو للمؤشر ، وبيانات الوقت والتاريخ والاوان يعاد تحويلها الى ارقام صحيحه للتعامل معها

إعلان البيانات :

يتم الاعلان عن انواع البيانات فى جزء التعبيرات والذى سنتناوله لاحقاً

مثال

```
int   (bool,color,datetime);
double;
string;
```

قبل بدء التشغيل (ماعدا فى جزء المهمات) فقد أُعلن عن الاهمية القصوى لتلك البيانات

مثال

```
int    i = 1 / 2;     // no types casting, the result is 0
int    i = 1 / 2.0;   // the expression is cast to the double type, then is to the target type of int, the result is 0
double d = 1.0 / 2.0; // no types casting, the result is 0.5
double d = 1 / 2.0;   // the expression is cast to the double type that is the same as the target type, the result is 0.5
double d = 1 / 2;     // the expression of the int type is cast to the target type of double, the result is 0.0
string s = 1.0 / 8;   // the expression is cast to the double type, then is to the target type of string, the result is "0.12500000" (the string containing 10 c
string s = NULL;      // the constant of type int is cast to the target type of string, the result is "0" (the string containing 1 character)
string s = "Ticket #"+12345; // the expression is cast to the string type that is the same as the target type, the result is "Ticket #12345"
```

اعلان البيانات يتم فقط مع الثوابت وليس مع المتغيرات

ثوابت الاعداد :

وتنقسم الى

الارقام العشرية : وهى الارقام من **0** الى **9** ولا يمكن اطلاقاَ أن نبدأ بالــ صفر

مثال :

```
12, 111, -956 1007
```

الارقام السداسية العشرية : وهى الارقام من **0** الى **9** والحروف من **a** الى **f** أو من **A** الى **F** وذلك لكتابة القيم من **10** الى **15** وتكتب هكذا **0X** أو **0x**

مثال :

```
0x0A, 0x12, 0X12, 0x2f, 0xA3, 0Xa3, 0X7C7
```

المدى المسموح به من الارقام هو من **-2147483648** وحتى **2147483648**

اى رقم يتجاوز هذا المدى تصبح النتيجة غير معرفه

ثوابت الحروف :

اى حرف مفرد ذُكر فى عملية من العمليات البرمجيه او بيان سداسى عشرى (ذكر مسبقاً) ويحتوى على حروف بالشكل '\x10' يسمى ثابت حرفى من النوع الصحيح
هناك حروف اخرى مثل الاقتباس المفرد (') والاقتباس المزدوج (") وعلامة الاستفهام (?)
والشرطة المائلة (\)

بالاضافه ان أحرف التحكم يمكن ان تكتب مجمّعه مبدوءه بشرطة مائله طبقاً للجدول التالى

line feed	NL (LF)	\n
horizontal tab	HT	\t
carriage return	CR	\r
reverse slash	\	\\
single quote	'	\'
double quote	"	\"
hexadecimal ASCII-code	hh	\xhh

واذا كان الحرف مختلف عن الجدول السابق ، ستصبح النتيجه غير معروفه

```
int a = 'A';
int b = '$';
int c = '©'; // code 0xA9
int d = '\xAE';   // symbol code ®
```

المدى المسموح به للثوابت الحرفيه هو من صفر وحتى 255

اى ثابت ياعدى هذا النطاق تصبح النتيجه غير معروفه

القيم المنطقيه :

القيم المنطقيه تضمن فقط القيمتين **true or false** واللذان يمكن كتابتهم على شكل **1 or 0** على التوالى

قد تكتب **True or TRUE** وكذلك **False or FALSE**

```
bool a = true;
bool b = false;
bool c = 1;
```

القيم المنطقيه فقط لا تمثل الا بالرقمين صفر و **1**

ثوابت النقطه العائمه (الارقام بالفاصله العشرية) :

وهى تتكون من جزء رقم صحيح ثم علامة عشرية (.) ثم الكسر

الثابت والكسر كلاهما يمثلان ارقاماً عشريه ثابته

مثال :

```
double a = 12.111;
double b = -956.1007;
double c = 0.0001;
double d = 16;
```

ثوابت النقطة العائمة (الارقام بالفصلة العشرية) لا يمكن ان تتجاوز قيمتهم المدى من

-1.7 * e-308 وحتى 1.7 * e308

اى نتيجه خارج هذا المدى تصبح النتيجه غير معروفة

ثوابت السلسة النصيه الحرفيه :

هى سلسلة من الحروف بتكويد -ASCII تحتوى على المعرفين "Character constant"
فهى مصفوفه من الحروف الموجوده بالاقواس من نوع البيانات المتسلسلة ، وعند الحاجه الى ادخال اقواس مزدوجه للتعريف (") داخل السلسلة نضع علامة شرطة مائلة معكوسة قبلها
طول السلسلة يقع فى المدى بين 0 وحتى 255 حرف ، وعند تجاوز هذا المدى سيتم رفض الحروف الزائده على اليمين

مثال :

```
"This is a character string"
"Copyright symbol \t\xA9"
"this line contains a line feed symbol \n"
"C:\\Program Files\\MetaTrader 4"
"A" "1234567890" "0" "$"
```

ثوابت الالوان :

يمكن تمثيل الثوابت اللونية بثلاث طرق : حرفياً أو بأعداد صحيحه أو باسم اللون نفسه (بالاسم المتعارف عليه طبقاً لجدول الالوان البرمجيه)

التمثيل الحرفي : يتكون من ثلاثة اجزاء كل جزء يوضح القيمة العدديه لتوزيع الالوان الرئيسية التى تتكون منها كل الالوان وهى (الاحمر والاخضر والازرق)
ويتم تمثيلهم بالطريقة الاتيه : اولاً يبدا الثابت بالحرف C ثم يحتوى على قوس داخله القيمه اللونيه العدديه والذى تتراوح قيمته من 0 وحتى 255

التمثيل بأعداد صحيحه : ويكتب بطريقة ارقام عشرية أو سداسية عشرية (سبق الاشارة اليها)
السداسية العشرية مثل : 0x00BBGGRR

حيث RR تشير الى نسبة اللون الاحمر
وحيث GG تشير الى نسبة اللون الاخضر
وحيث BB تشير الى اللون الازرق

التمثيل باسم اللون نفسة : ويكون طبقاً للاسم المتعارف عليه طبقاً لجدول الالوان البرمجيه

مثال :

```
// symbol constants
C'128,128,128'    // gray
C'0x00,0x00,0xFF' // blue
// named color
Red
Yellow
Black
// integer-valued representation
0xFFFFFF          // white
16777215          // white
0x008000          // green
32768             // green
```

ثوابت الوقت والتاريخ :

ثوابت الوقت والتاريخ يمكن تمثيلها كسطر حرفى يتكون من 6 أجزاء

وتلك الاجزاء عبارة عن (السنة ، الشهر ، التاريخ ، الساعات ، الدقائق ، الثوانى)

الثابت يحتوى على قوس واحد فقط ويبدأ بالحرف D

يمكن تجاوز (year, month, date) أو (hours, minutes, seconds) او كلاهما

ثوابت الوقت والتاريخ لا يجب ان تتجاوز المدى الزمنى من عام 1970 وحتى 31 ديسمبر عام 2037

مثال :

```
D'2004.01.01 00:00'     // New Year
D'1980.07.19 12:30:27'
D'19.07.1980 12:30:27'
D'19.07.1980 12'        //equal to D'1980.07.19 12:00:00'
D'01.01.2004'           //equal to D'01.01.2004 00:00:00'
D'12:30:27'             //equal to D'[compilation date] 12:30:27'
D''                     //equal to D'[compilation date] 00:00:00'
```

يُتبَع ،