

Chapter one

Data Variables and constants

- 1- When you input data into a computer, the computer stores it in its memory
- 2- Computer's memory contains millions of little boxes "Bytes"
- 3- The number of memory bytes represents memory size
- 4- All high level programming languages including Visual Basic allow you to set up locations in memory and give each group of locations a name

البيانات الثابتة والمتغيرة :-

- اي بيانات تخزن في ذاكرة الحاسب الداخلية
- تتكون الذاكرة الداخلية من ملايين الخلايا المتساوية وتسمى كل خلية (Byte)
- عدد خلايا الذاكرة يمثل حجم الذاكرة

نلاحظ أن :- كلما زاد عدد الخلايا كلما كان حجم الذاكرة أكبر كلما زادت السعة التخزينية

- تسمح كل لغات البرمجة كل لغات البرمجة باعطاء اسماء للخلايا التي سيتم تخزين البيانات بها حتي يسهل تتبعها والتعامل معها

Variables:-

Data can be changed during program Execution

المتغيرات :- أسماء الخلايا التي تشغلها القيم المخزنة والتي تتغير أثناء تشغيل البرنامج

Constants:-

Data cannot be changed during program Execution

Like:- → $\pi = 3.14$

الثوابت :- أسماء الخلايا التي تشغلها القيم المخزنة والتي لا تتغير أثناء تشغيل البرنامج

Byte :- The size of memory needed by the variable of this type

Range :- refers to the largest and smallest values that can be stored in a numeric variable of the given type.

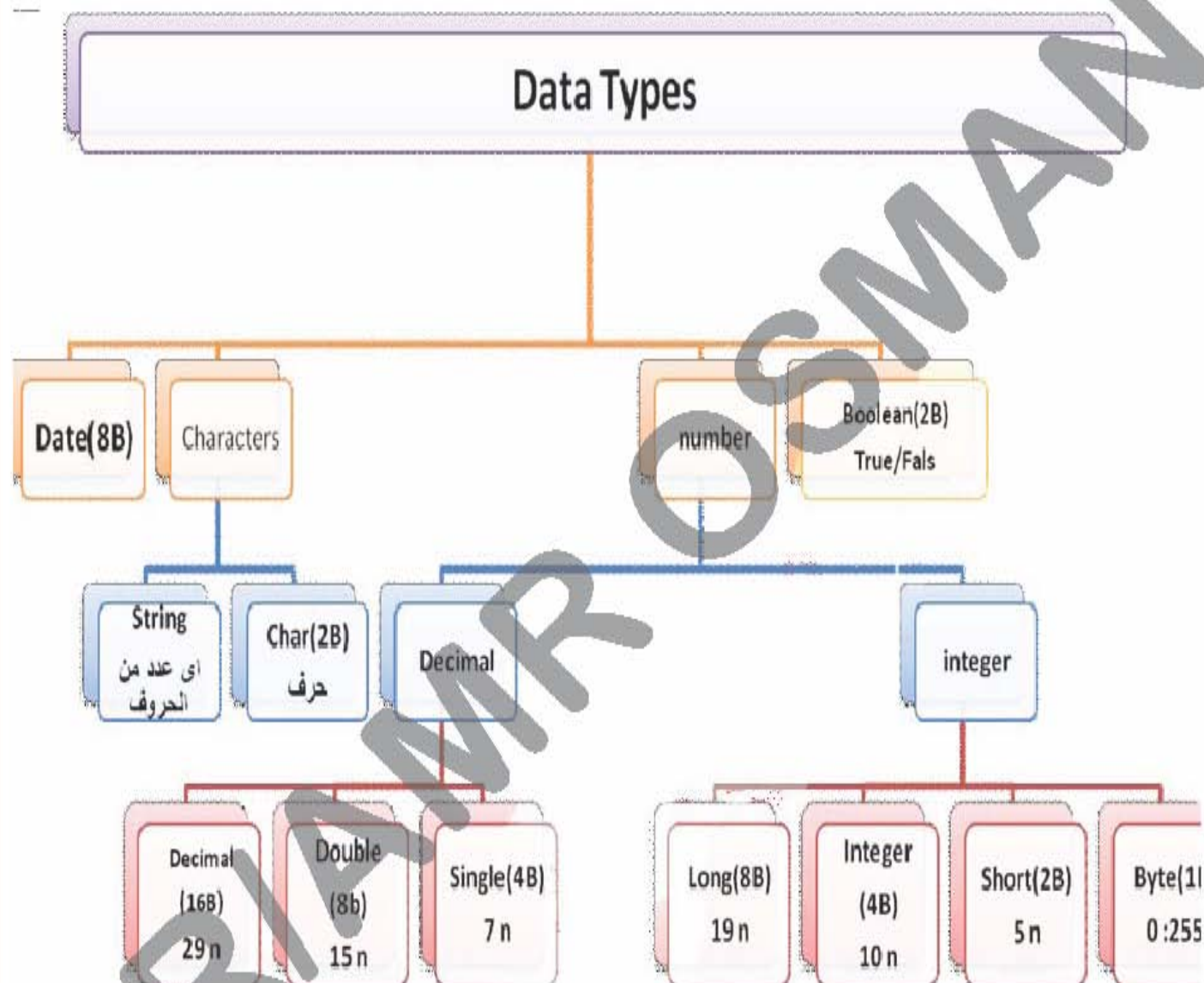
Precision :- Different numeric data types have different precision.

البايت (Byte) :- يقصد به حجم الذاكرة التي يحتاجها المتغير من هذا النوع .

المدى (Range) :- هو حدود القيم المتاحة للمتغير .

الدقة (Precision) :- هي عدد الخانات التي يتم تمثيل المتغير بها "خاصة بالمتغيرات الرقمية"

أنواع البيانات التي يتم تخزينها



رسم توضيحي لأنواع البيانات

إزاي أعرف أحدد عدد الخلايا للمتغيرات الرقمية بسهولة :

د الذاكرة إما ١ أو ٢ أو ٤ أو ٨ أو ١٦ بايت

٢ نقوم بقسمة الدقة على ٢ ونقرنها لما هو مذكور في الخطوة رقم ١

مثال ١ : النوع integer الدقة ١٠ أرقام يبقى $10 / 2 = 5$ أقرب لل ٤ من ٨ يبقى عدد الخلايا ٤

مثال ٢ : النوع single الدقة ٧ أرقام يبقى $7 / 2 = 3.5$ أقرب لل ٤ من ٢ يبقى عدد الخلايا ٢

Naming rules in Visual Basic (Rules of naming the variables and constants)

- 1- Names must begin with a letter .
- 2- Names may consist of letters , digits , and underscores .
- 3- Names cannot contain any space or periods .
- 4- Names must not be reserved words. (such as Print , Dim ,Double , and Date)

أولاً : قواعد يجب الالتزام بها :-

- 1- يجب ان يبدأ الاسم بحرف من حروف الانجليزية
- 2- يأتي بعد الحرف الاول أى عدد من الحروف او الارقام
- 3- لايسمح ان يحتوى على رموز او علامات او مسافات ماعدا علامة ال Underscore
- 4- ممنوع استخدام الكلمات المحجوزة في VB.Net مثل " Color – Short – Double "

Rules of naming the program and facilitating its reading and not following them will not result in any errors

- 1-Identifiers should be meaningful.
- 2-Capitalize each successive word of the name.
- 3-The variable name should begin with three letters to show this variable type

For example :-

<u>Variable Name</u>	<u>Type</u>
intSalary	Integer
strStudentsNames	String
lngCount	Long

ثانياً : قواعد يستحسن الالتزام بها:

- 1- اختر اسماء ذات معنى مناسب .
- 2- فى الكلمات التى تتكون من مقطعين يمكن ان يبدأ كل مقطع بحرف Capital
- 3- يفضل ان يبدأ الاسم ب ٣ احرف تعطى انطباع عن نوع المتغير مثل intbookquantity

مثال للأمثلة الصحيحة :-

(BirthDay – First Name – PopulationOfEgypt –Spend_Mony)

مثال للأعداد الغير صحيحة مع بيان السبب :-

The following list shows some invalid variable names:

Variable Name	Cause of Invalidation of Variable Name
2ndWorldWar	Variable name must start with character
Spent.Money	Variable name contains period
Birth Day	Variable name contains space
Double	Variable name is a reserved word

Choosing the appropriate data type for a variable each variable

اختيار النوع المناسب لكل متغير :-

السبب	النوع المناسب	طبيعة البيانات المخزنة
أحسن استغلال للذاكرة	Boolean	True Or False
الاختيار الوحيد	String	مجموعة من الحروف والأرقام التي لن تدخل في حسابات كأرقام التليفونات مثلا
من الممكن ان يخزن الجنيهاات والقروش بدقة وبدون أخطاء التقريب كما انه سريع ف العمليات الحسابية	Decimal	مبلغ من المال
استهلاك قليل للذاكرة كما انه اسرع في الحسابات من المتغيرات الرقمية الأخرى	• Short	أعداد صحيحة :- • من -32768 الي 32767
استهلاك قليل للذاكرة كما انه اسرع من الأخرى	• Integer	• من -2147473648 الي 2147473647+
Long	• Long	• من -922337203685477508 الي 922337203685477508+
اسرع في العمليات الحسابية من Double	• Single • Decima	• أرقام بكسور عشرية :- من -10 ³⁸ الي +10 ³⁷

The constant

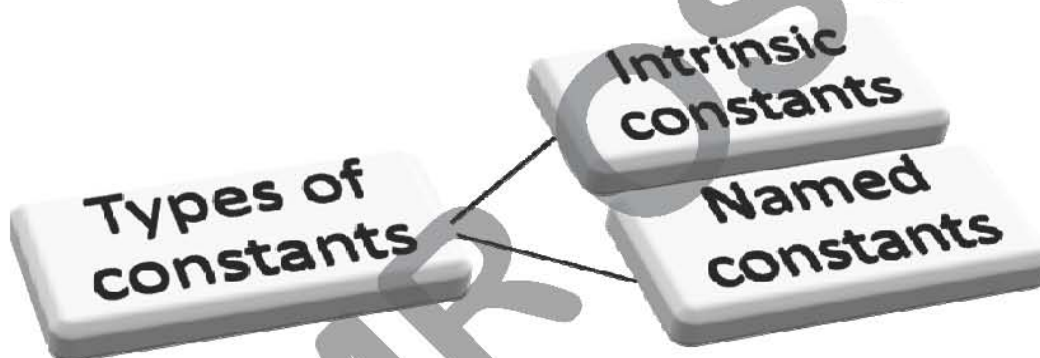
The constant is a store in the memory that contains unchangeable value during the running of the program

الثوابت :-

• تمثل الثوابت طريقة لاعطاء اسماء للقيم التي لا تتغير أثناء تشغيل البرنامج

Constants Types:-

- 1- Intrinsic constants
- 2- Named constants



أنواع الثوابت :-

1. ثوابت جوهريّة :- أي هي موجودة داخل برنامج VB.Net
2. ثوابت مسمّاة :- أي أنت الذي تقوم بإنشائها والإعلان عنها

الفرق بين النوعين :- ان الثوابت الجوهريّة :- لا تحتاج للإعلان عنها

لكن الثوابت المسمّاه :- لا بد من الإعلان هنا باستخدام Const

1- Intrinsic constants: - Are built into .Net Framework

For Example:-

(Color.black - Color.Blue - Color.red)

2-Named constants: - Are programmer – defined constants and are declared using the keyword “Const”

For Example: - `Const salary as integer = 20`

The Variables

The variable is a store in the memory that contains changeable value during the running of the program

Declaring a variable using dim statement:

Before using any variable, it should be declared first and this declaration will be in a certain form which is.

Dim Name of the variable **As** Variable type **=** the value

For example

- **Dim** ChildWeight **As** Integer
- **Dim** StudentName **As** String
- **Dim** UnitPrice **As** Decimal = 123.5
- **Dim** MyAddress **As** String = "102, Salah Salem, Cairo"

If you omit the optional equal sign and the initial value as in first three

examples above, the variable will be initialized with a default value as follows:

- If the variable was a numeric type, the default value is zero.
- If the variable was a string type, the default value is a

Zero-length (empty) string. ""

Try / Catch Block

A try/catch block: - is used to detect and handle exceptions (error) That are detected while the program is running.

Try / Catch Block

طريقة لاكتشاف الأخطاء التي تحدث أثناء تشغيل البرنامج والتعامل معها عن طريق إظهار رسالة تفيد بوجود خطأ والصيغة

Try

الجملة المحتمل مستخدم البرنامج ان يخطئ فيها

Catch

الرسالة التي تظهر لمستخدم البرنامج موضح بها الاخطاء التي حدثت

End try**F.ex :-****try**

```
Dim x,y as integer
Dim z as double
X=5
Y=2
Z = x / y
Msgbox("the result is " & z)
```

إذا كان الكود صحيح
سيقوم بطباعة الناتج

اما

لو كان الكود به خطأ فسيتم
تطبيق جملة

Catch**Catch**

```
Msgbox(" divided by zero ")
```

End try

الناتج هنا

The result is 2.5

F.ex :-

try

```

Dim x,y as integer
Dim z as double
X=5
Y=0
Z = x / y
Msgbox("the result is " & z)

```

إذا كان الكود صحيح
سيقوم بطباعة الناتج

أما

Catch

```
Msgbox("divided by zero ")
```

لو كان الكود به خطأ فسيتم
تطبيق جملة

End try

Catch

الناتج هنا

The result is divided by zero

Calculations

((العمليات الحسابية))

operator	Operation	Examples
+	Addition	5+3=8
-	Subtaction	5-3=2
*	Multiplication	5*3=15
/	Division	5/2=3.5
\	Integer division	5\3=2
Mod	Modullus	5mod2=1
^	exponentiations	5^2=25

Precedence of operations

- 1- ()
 - 2- ^
 - 3- * /
 - 4- \
 - 5- Mod
 - 6- + -
-

Q5- what is the result of the following arithmetic operations when:

(X = 3, Y = 2, N = 5)

1. $X * (Y + 3)$
2. $X * (N + 1) ^ 2 * Y ^ 2$
3. $Y ^ X * X$

.....

.....

.....

.....

.....

.....

.....

.....