



جامعة الزقازيق – كلية الهندسة – قسم هندسة الحاسبات والمنظومات



Microsoft
Visual Basic 6.0

CSE100 الحاسبات والبرمجة ٢

د/ عمرو أحمد زامل

https://dramrzamel.github.io/CSE100_VB6/

or

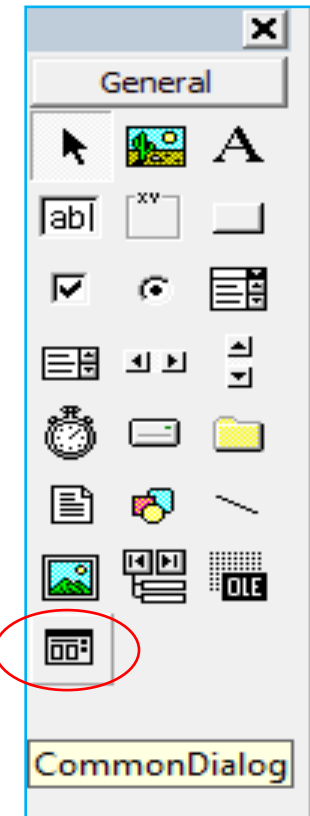
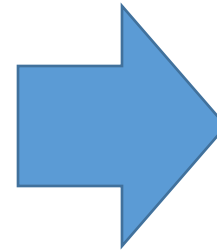
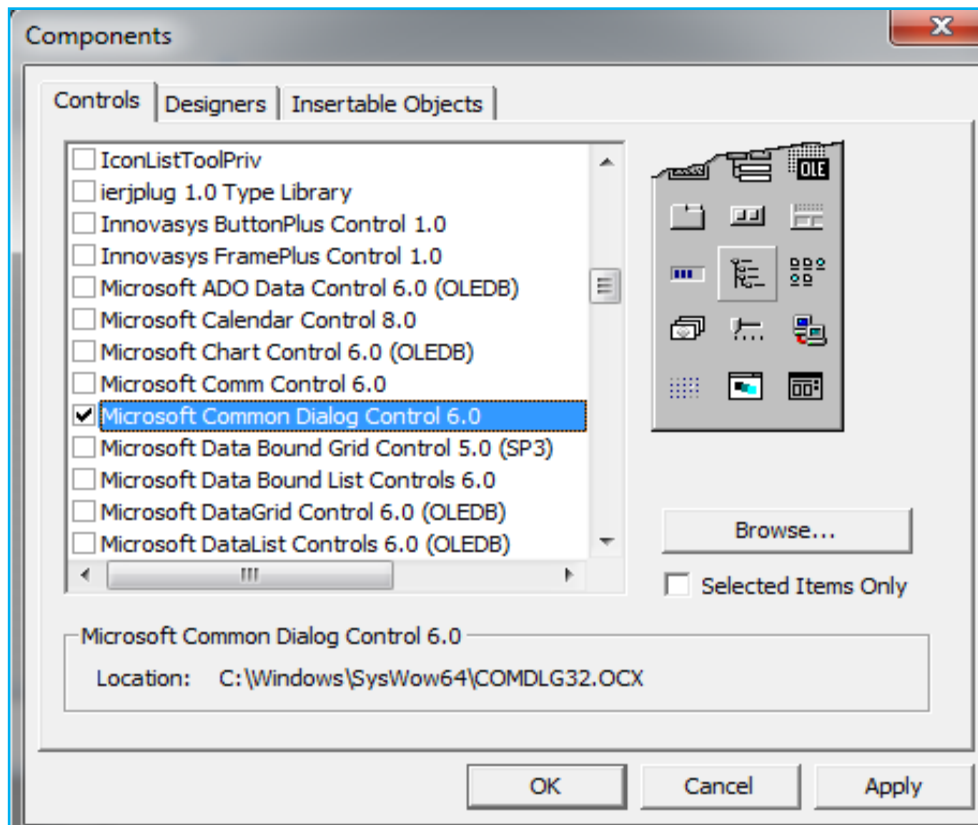
bit.ly/CSEVB6

المحاضرة ٦ : أساسيات البرمجة في الفيجول بيسيك

- التعامل مع الكيانات الهامة.
- بناء شريط القوائم وشريط الأدوات.
- بناء صناديق الحوار والرسائل.

صناديق الحوار المدمجة Built-in Dialog Boxes:

- مثل صناديق اختيار أشكال الحروف Fonts أو الألوان Colors و التحكم في الطباعة.
- يتم اضافة عنصر جديد لأدوات المشروع كالتالي:



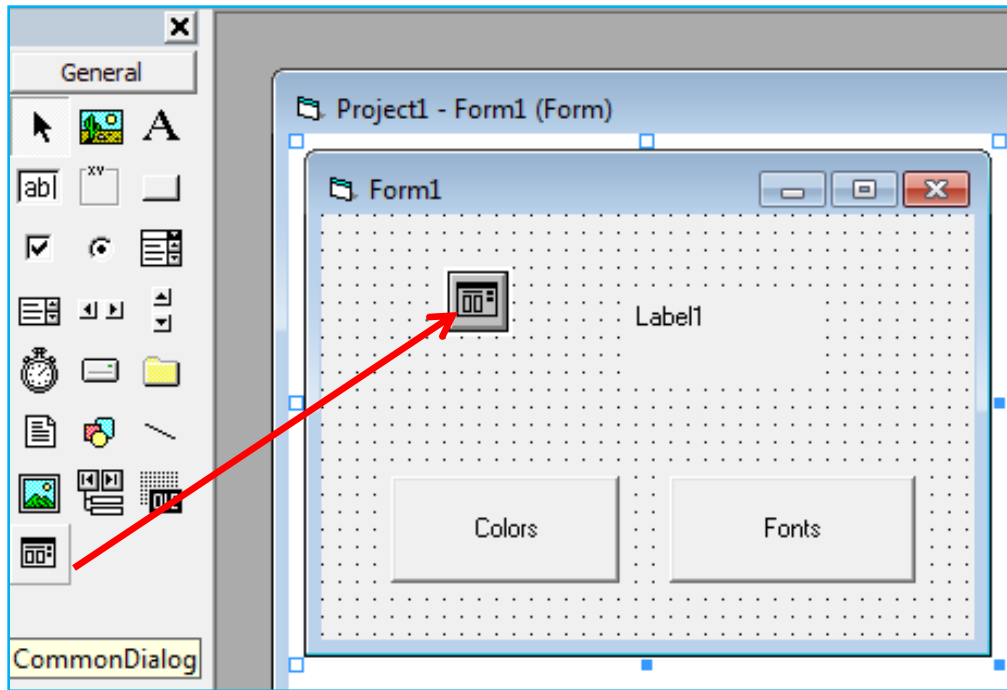
صناديق حوار الألوان Colors وأشكال الحروف Fonts:

- فتح صندوق الألوان: نكتب اسم متحكم ال Common Dialog ثم مؤثر النقطة ثم `ShowColor` مثل: `cdTest.ShowColor` حيث أن `cdTest` هي الاسم الكودي للمتحكم.
 - يمكن الوصول الى اللون الذي تم اختياره من صندوق الألوان بالأمر التالي: `cdTest.Color`
 - ويتم التعامل مع صندوق أشكال الحروف Font بنفس الكيفية:
- `cdTest.ShowFont` فتح الصندوق
 - `cdTest.FontSize` الوصول لحجم الخط الذي تم اختياره

صناديق حوار الألوان Colors وأشكال الحروف Fonts:

■ **مثال:** التحكم في مظهر لافته مثل لون الخلفية و شكل الخط عن طريق صناديق الألوان و

أشكال الحروف:



■ خاصية Name لمتحكم صندوق

الحوار الخاص تحمل cdTest.

صناديق حوار الألوان Colors وأشكال الحروف Fonts:

■ **مثال:** التحكم في مظهر لافته مثل لون الخلفية و شكل الخط عن طريق صناديق الألوان و

أشكال الحروف:

```
Private Sub CmdColors_Click()  
cdTest.ShowColor  
Label1.BackColor = cdTest.Color  
End Sub
```

```
Private Sub CmdFonts_Click()  
cdTest.ShowFont  
Label1.Font.Size = cdTest.FontSize  
Label1.FontName = cdTest.FontName  
Label1.FontBold = cdTest.FontBold  
End Sub
```

■ الكود: التحكم في بعض خصائص

الخط مثل الحجم، نوعية الخط و

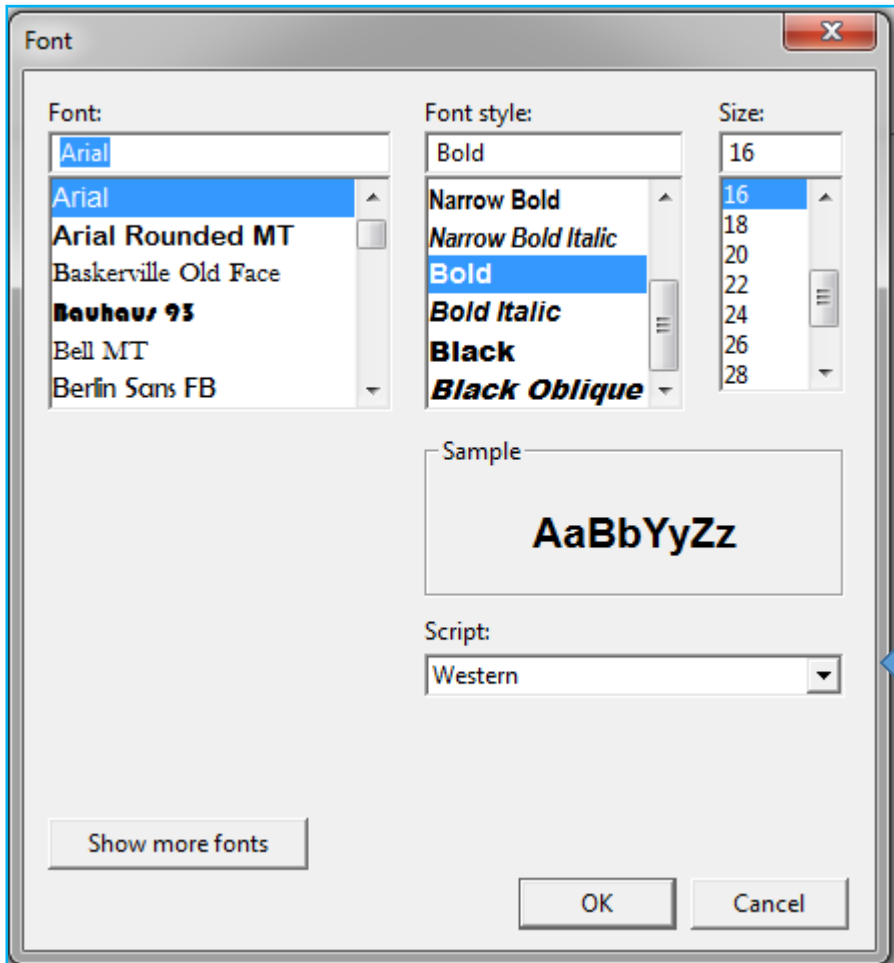
كونه سميكاً Bold أم لا.

صناديق حوار الألوان Colors وأشكال الحروف Fonts:

■ **مثال:** التحكم في مظهر لافته مثل لون الخلفية و شكل الخط عن طريق صناديق الألوان و

أشكال الحروف:

■ وضع التنفيذ:



صناديق حوار الملفات Files والطابعة Printer:

■ فتح ملف:

cdTest.ShowOpen

■ ويمكن الوصول لمسار الملف المختار عن طريق

cdTest.FileName

■ حفظ ملف:

cdTest.ShowSave

■ عرض صندوق الطابعة:

cdTest.ShowPrinter

- المتغيرات Variables:

- الاعلان عن المتغيرات.

- تسمية المتغيرات وأنواعها.

- نطاق استخدام المتغيرات Variables Scope.

- الثوابت Constants.

- المجموعات المترابطة من المتغيرات Arrays

- ايعازات البرنامج: العمليات الحسابية

- اتخاذ القرار: ايعاز If

- **المتغير Variable**: هو عبارة عن اسم موقع تخزين Storage location في الذاكرة الرئيسية Main Memory والذي يمكن أن تتغير قيمة محتوياته أثناء تنفيذ البرنامج.
- التعامل معه (قراءة محتوياته أو تخزين بيانات فيه) عن طريق اسم معين.
- نحتاج الى المتغيرات لكي نتمكن من تنفيذ وظائف مختلفة في البرنامج مثل اجراء عمليات حسابية ، ادخال / اخراج بيانات من / إلي البرنامج ، ...
- أي **متغير** له ثلاثة خصائص: **الاسم** Variable name ، **القيمة** و **النوع** Data Type.

الاعلان عن المتغيرات Variables Declaration :

الصورة العامة للاعلان عن متغير:

Dim Variable Name [**As** Variable Type]

اسم المتغير

نوع المتغير

مثال :

Dim std **As** String

قواعد تسمية المتغيرات:

- لابد أن يبدأ بحرف .
- لا يسمح بالمسافات ولكن ممكن يحتوي على أرقام أو (_)underscores.
- أقصى عدد من الحروف للاسم هو 255 حرف.
- لايسمح باستخدام أي من ال Keywords المعروفة في ال VB مثل **sub**, **End**, **If**,
- لا يهم كون الحروف كبيرة أو صغيرة بمعنى أن **sText** , **stext** يمثلان نفس الاسم.

أمثلة لأخطاء في الاعلان عن متغيرات:

الاعلان	الخطأ
Dim My Name As String	المسافة
Dim My.Name As string	النقطة Dot
Dim (Item#1) As Integer	الرموز #, (,)
Dim 1st As date	الاسم يبدأ برقم
Dim sMsg As string Dim smsg	تكرار نفس اسم المتغير مرتين في ايعازين مختلفين
Dim End As String	استخدام أحد ال keywords وهي كلمة End

أنواع المتغيرات : Data Types

متغيرات مركبة	المتغيرات الأساسية 11 نوع
تسمى User-defined variables	Boolean, Byte, Currency, Date, Single, Double, Integer , Long, Object, String, and Variant.

أنواع المتغيرات Data Types :

أهم الأنواع التي نتعامل معها:

نوع المتغير	Prefix	مثال للاسم	مثال للقيمة	القيمة الغيائية
String	s	sText	"Hello"	empty
Integer	n	nSid	415209	0
Single	f	fSalary	1200.66	0
Boolean	b	bState	True / False	False

أنواع المتغيرات : Data Types

نوع القيمة	نوع المتغير	مساحة التخزين (Byte)	Range
قيم صحيحة	Integer	2	-32,768 to 32,767
	Long	4	-2,147,483,648 to 2,147,483,647
قيم نسبية floating point	Single حتى 7 أرقام عشرية	4	$\pm 1.40129 \times 10^{-45}$ to $\pm 3.40283 \times 10^{38}$
	Double حتى 14 رقم عشري	8	$\pm 4.94065 \times 10^{-324}$ to $\pm 1.7976 \times 10^{308}$
سلاسل حرفية	String	1 per character	Up to 65,400 characters
قيم عامة Numeric/string	Variant	16	As Double for numeric

أنواع المتغيرات : Data Types

نوع القيمة	نوع المتغير	مساحة التخزين (Byte)	Range
True / False	Boolean	2	True or False
للتعامل مع العملات	Currency	4	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
التاريخ	Date	8	January 1, 100 to December 31, 9999
قيم عامة	Byte	1	0 to 255 (unsigned numbers)
أي تشكيله من الأنواع المذكورة سابقا	User-defined	حسب ما يحتاجه كل عنصر	كل عنصر له المدى الخاص به

النوعية المركبة User-defined:

الاجراء العام



```
(General) (Declarations)

Private Type Point
    x As Integer
    y As Single
End Type
```

■ الاعلان عن النوعية **Point**:

■ الاعلان عن متغير **MyPoint** من النوعية **Point**:

```
Private Sub Command1_Click()
    Dim MyPoint As Point
    MyPoint.x = 3
    MyPoint.y = 5.2548
End Sub
```

■ الوصول لعناصر المتغير **MyPoint**:

■ يمكن استخدام النوعية **Point** داخل أي اجراء آخر في هذا النموذج.

إذا لم يتم ذكر نوع المتغير في ايعاز الاعلان عنه فان ال VB يعتبره من النوعية **Variant** و عليه فان اليعازين التاليين متكافئين تماما:

Dim Var1

Dim Var1 As Variant

يتم اعطاء قيمه للمتغير باستخدام مؤثر التخصيص = ، أمثلة:

```
Dim bValue As Boolean
bValue = True
```

```
Dim sName As String
sName = "Ali"
```

```
Dim sName As String
sName = Inputbox ("Enter your name")
```

```
Dim nAge As Integer
nAge = Val(Text1.text)
```

طريقة أخرى للإعلان عن متغير:

وتسمى الاعلان الضمني Implicit Declaration: نستخدم حرف خاص في آخر اسم المتغير يدل على نوعيته.

النوعية	Suffix	مثال
Integer	%	Dim nVar1%
Single	!	Dim fVar2!
String	\$	Dim sTemp\$
Long	&	Dim sVar3&
Double	#	Dim dVar4#
Currency	@	Dim cVar5@
Boolean, Byte, Date	None	Dim bVar6,X,Y

يمكن الاستغناء عن كلمة **Dim** في الاعلان الضمني بشرط تخصيص قيمة ابتدائية

nVar1% = 100	fVar! = 50.236	X = True	للمتغير:
--------------	----------------	----------	----------

نطاق استخدام المتغيرات Variables Scope:

المتغير يمكن أن يكون متاحا في كل اجراء داخل البرنامج أو مقصورا على اجراء معين بناء على **طريقة** و **موقع** الاعلان عنه.

النوع	طريقة الاعلان (Keyword)	موقع الاعلان	Availability
عام Public	Public	General Declaration حيث الاعلان العام	متاح لأي اجراء داخل أي نموذج في البرنامج
محلي Local	Private / Dim	داخل اجراء معين	داخل هذا الاجراء فقط

الاعلان عن متغير عام `Public sUserName As String`

الوصول اليه داخل اجراء نموذج `FrmRegister.sUserName = "Hassan"`

المتغير الساكن والثابت:

المتغير الساكن Static Variable

عند الانتهاء من تنفيذ اجراء ما فانه يتم تصفية المتغيرات الخاصة (ازالتها من الذاكرة) بهذا الاجراء ولكن اذا اردنا الاحتفاظ بقيمة المتغير لحين تنفيذ الاجراء مرة أخرى فيلزم الاعلان عن كونه ساكنا:

Static nPageNumber As Integer

الثوابت Constants

عند استخدام قيمة ثابتة أكثر من مرة في البرنامج يفضل أن نستخدمها كثابت ونعلن عنها **Const**، مثل:

Const PI = 3.14159265359

Const g = 9.81

يمكن أن يكون الثابت عمومياً **Public** أو خاص بإجراء معين.

المجموعات المتراسة Arrays:

□ للتعامل مع عدد معين من المتغيرات من نفس النوعية:

Dim ArrayName (1 To n) As Type

اسم المجموعة المتراسة

عدد عناصرها n

نوعية العناصر

■ هنا يتم حجز عدد من مواقع الذاكرة تحت اسم **ArrayName** وكل عنصر له مساحة

تخزينية حسب نوعية العناصر **Type**.

■ موضع العنصر **Index** داخل المجموعة يبدأ من **1** حتي **n** بمعنى أن **ArrayName(1)** تشير

إلى العنصر **الأول**، **ArrayName(3)** تشير إلى العنصر **الثالث** وهكذا.

□ مثال:

```
Private Sub Command1_Click()  
Dim nArray1(1 To 5) As Integer  
'Assign value for the first element  
nArray1(1) = 10  
'Receive value for the third element  
nArray1(3) = Val(Text1.Text)  
'Print the value of the last element  
Print nArray1(5)  
End Sub
```

■ لا حظ السطور التي تظهر باللون الأخضر في الكود هي تعليقات أو ملاحظات Remarks و

تكون مسبوقة ب (').

■ في المثال السابق: عند كتابة العنصر الخامس على النموذج فانه يتم كتابة 0.

العمليات الحسابية :Arithmetic Operations

رموز العمليات الأساسية :

```
Dim x As Integer, y As Integer, z  
As Single
```

```
x = 13
```

```
y = 2
```

```
z = x / y
```

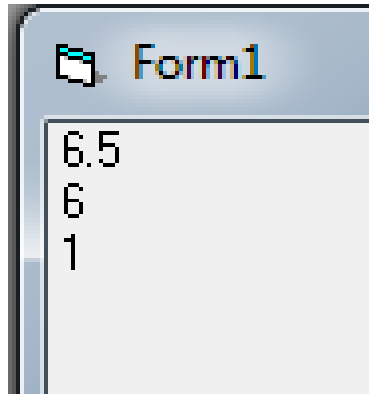
```
Print z
```

```
z = x \ y
```

```
Print z
```

```
z = x Mod y
```

```
Print z
```



الرمز	العملية
+	الجمع
-	الطرح
*	الضرب
/	القسمة العادية
\	القسمة الصحيحة
^	الرفع لأس
Mod	المتبقي من القسمة

□ **ترتيب العمليات الحسابية**: الأقواس لها الأولوية المطلقة ثم الرفع لأس ثم الضرب و

القسمة ثم القسمة الصحيحة ثم المتبقي Mod و أخيرا الجمع و الطرح.

■ لكتابة عدد مثل 2×10^3 نستخدم حرف E (متغير Single) أو D (متغير Double) كالتالي:

$X = 5\text{E}3$	$X = 5000$
$X = 7\text{E}-2$	$X = 0.07$
$X = 23.6489\text{D}3$	$X = 23648.9$

■ ولاحظ أنه بمجرد كتابة هذا الايعاز داخل الكود فان ال **VB** يحول العدد تلقائيا الي الصورة

العادية (العمود الأيمن).

صمم برنامجا يحسب جذور المعادلة التربيعية:

$$aX^2 + bX + c = 0$$

رتب البرنامج بحيث يتم ادخال قيم المعاملات a,b,c عن طريق ثلاثة صناديق نصوص،

ويتم عرض النتيجة على لافتات. استبعد الحلول التخيلية واتخذ الاجراءات اللازمة

لحماية البرنامج من أخطاء التشغيل مثل القسمة على صفر أو التعامل مع قيمة غير

عددية.

• من المعلوم أن الجذور التربيعية يتم حسابها كالتالي:

$$X_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

حل المثال 1: وضع التصميم:

Form1

حل معادلة من الدرجة الثانية

a=

b=

c=

X1 =

X2 =

Calculate

Exit

```

Private Sub CmdCalculate_Click()
Dim a As Single, b As Single, c As Single
Dim X1 As Single, X2 As Single, Y As Single
'avoid division by 0
If Val(Txta.Text) = 0 Then
    MsgBox "Invalid value for a"
End If
'receiving a,b,c
a = Val(Txta.Text)
b = Val(Txtb.Text)
c = Val(Txte.Text)
'avoid imaginay solutions
Y = b ^ 2 - 4 * a * c
If Y >= 0 Then
    X1 = (-b + Sqr(Y)) / (2 * a)
    X2 = (-b - Sqr(Y)) / (2 * a)
    LblX1.Caption = X1
    LblX2.Caption = X2
Else
    MsgBox "Imaginary Solution"
End If
End Sub

```

```

Private Sub CmdExit_Click()
End
End Sub

```

وضع التنفيذ:

Form1

حل معادلة من الدرجة الثانية

a= 2

b= 2

c= 3

X1 =

X2 =

Calculate

Exit

SecOrder_EqProje... X

Imaginary Solution

OK

Form1

حل معادلة من الدرجة الثانية

a= -2

b= 5

c= 1

X1 = -0.1861407

X2 = 2.686141

Calculate

Exit

- كيف يمكن تعديل حل المثال السابق بحيث اذا كان الجذران $X1$ و $X2$ متساويين يتم كتابة النتيجة على لافتة واحدة تحت اسم $X1,2$ بينما تختفي اللافتة الأخرى.

Form1

حل معادلة من الدرجة الثانية

a= 1

b= 2

c= 1

X1,2= -1

Calculate

Exit

- تم استخدام الدالة **Sqr()** لحساب الجذر التربيعي وهي أحد الدوال الخاصة المتاحة في ال **VB** كما أن هناك دوال أخرى رياضية وهندسية هامة مثل:

function	القيمة التي ترجعها الدالة
RND	توليد رقم عشوائي بين 0,1
ABS(X)	القيمة المطلقة $ X $, Absolute of X
EXP(X)	e^X , $e=2.71828$
LOG(X)	Natural logarithm: $\ln(X)$
Sin (x)	Trigonometric functions
Cos(x)	
Tan(x)	
FIX(X)	الجزء الصحيح من X

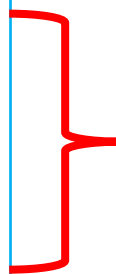
- ال VB لا يحتوى على نوعية متغيرات تتعامل مع الأعداد التخيلية أو المركبة **Complex numbers** ولذلك اذا أردنا التعامل مع هذه الاعداد لابد أن يكتب المستخدم بنفسه دوال أو أجزاء من الكود تساعد في ذلك.
- تابع مثال 1: يمكن تعديل البرنامج بحيث يتعامل مع أي مدخلات و يحسب الجذور التخيلية اذا كانت ضمن الحل.
- العدد التخيلي يكتب كالتالي: $X = R \pm i M$ حيث تمثل **R** الجزء الحقيقي بينما **M** تمثل الجزء التخيلي و **i** هي الجذر التربيعي ل **-1** كما هو معروف.

■ عند حساب جذور المعادلة التربيعية:

$$R = -\frac{b}{2a}, M = \frac{\sqrt{4ac - b^2}}{2a}$$

تابع مثال 1: التعامل مع الاعداد التخيلية

```
Private Sub CmdCalculate_Click()  
Dim a As Single, b As Single, c As Single  
Dim X1 As Single, X2 As Single, Y As Single  
Dim R As Single, M As Single  
'avoid division by 0  
If Val(Txta.Text) = 0 Then  
    MsgBox "Invalid value for a"  
End If  
'receiving a,b,c  
a = Val(Txta.Text)  
b = Val(Txtb.Text)  
c = Val(Txte.Text)  
'avoid imaginay solutions  
Y = b ^ 2 - 4 * a * c  
If Y >= 0 Then  
    X1 = (-b + Sqr(Y)) / (2 * a)  
    X2 = (-b - Sqr(Y)) / (2 * a)  
    LblX1.Caption = X1  
    LblX2.Caption = X2  
Else  
    'MsgBox "Imaginary Solution, Y is negative"  
    R = -b / (2 * a)  
    M = Sqr(-Y) / (2 * a)  
    LblX1.Caption = R & " + i " & M  
    LblX2.Caption = R & " - i " & M  
End If  
End Sub
```



اتخاذ القرار: ايعاز If

If Condition Then Statement

١- ايعاز السطر الواحد:

- حيث أنه اذا تحقق الشرط **Condition** فانه يتم تنفيذ ايعاز **Statement** و بعد ذلك يتم الانتقال الى السطر التالي في الكود. اذا لم يتحقق الشرط يتم الانتقال الى السطر التالي في الكود مباشرة.
- يتم السؤال في ال Condition باستخدام المؤثرات الثنائية والمنطقية:

المعنى	المؤثر الثنائي
هل X أكبر من Y	$X > Y$
أقل من	$<$
أكبر من أو يساوي	$>=$
أقل من أو يساوي	$<=$
يساوي	$=$
لايساوي	$<>$

المعنى	المؤثر المنطقي
لابد أن يكون كل الشروط محققة	AND
يكفي أن يكون أحد الشروط محقق	OR
النفي	NOT

ما الفرق بين عمل كل من AND و OR؟

اتخاذ القرار: ايعاز If

If Condition **Then**

Statement1

...

Else

Statement1

...

End If

If Condition1 **Then**

Statement1

...

ElseIf Condition2 **Then**

Statement1

...

Else

Statement1

...

End If

3- ايعاز If-Else

2- ايعاز If متعدد الأسطر

If Condition **Then**

Statement1

Statement2

...

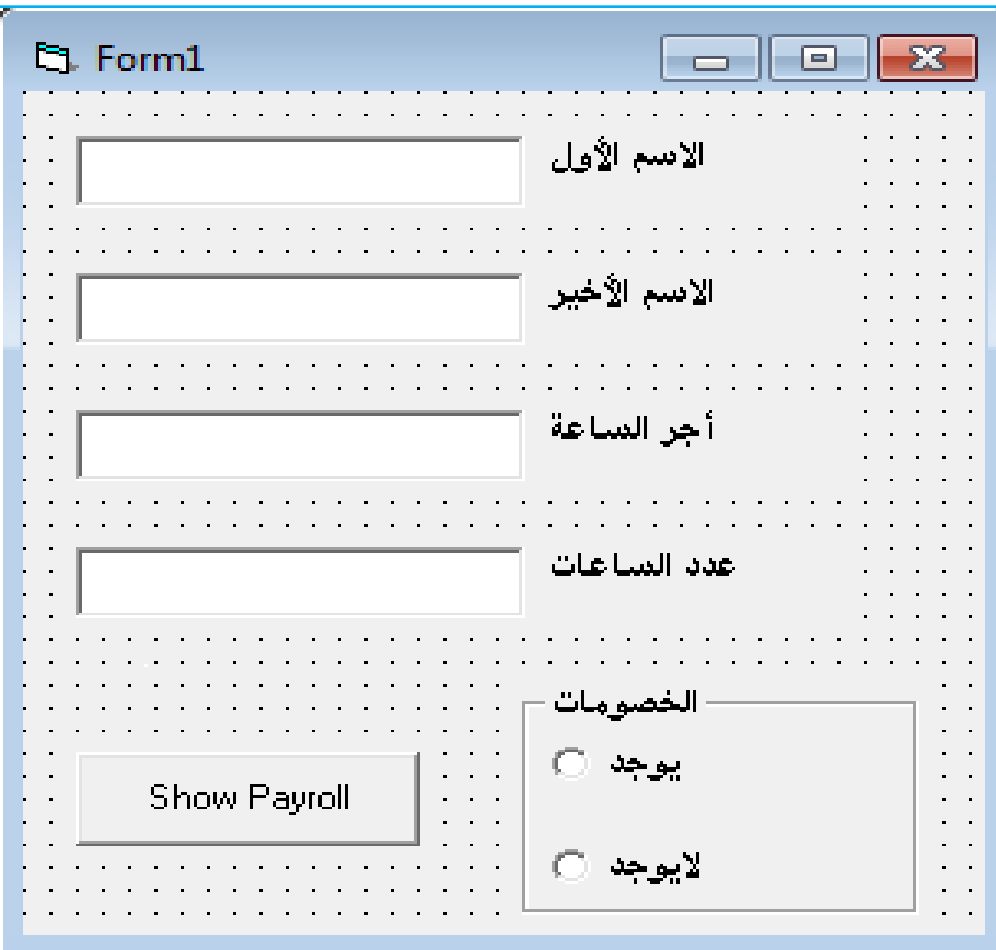
End If

صمم برنامجا يستقبل بيانات موظف مثل الاسم الأول و الاسم الأخير و أجر الساعة و عدد ساعات العمل بحيث يعرض اسم العامل كاملا علي نموذج آخر مضافا اليها الأجر الصافي الذي يتم حسابه كالتالي:

الأجر الصافي = أجر الساعة * عدد الساعات – الخصومات

بحيث اذا كان أجر الساعة بين 100 و 90 يكون الخصم 15% من الأجر قبل الخصم أما اذا كان أجر الساعة أكبر من 75 وأقل من 90 فان الخصم يكون 10%، واذا كان أكبر من 50 وأقل من 75 فان الخصم يكون 8% و أخيرا اذا كان الأجر أقل من 50 فان الخصم يكون 5% وقد لا يوجد خصومات. رتب البرنامج بحيث يتم اعلان متغير مركب يضم كل بيانات الموظف.

حل مثال 2: وضع التصميم:



Form1

الاسم الأول

الاسم الأخير

أجر الساعة

عدد الساعات

الخصومات

☐ يوجد

☐ لا يوجد

Show Payroll



Form2

اسم الموظف

Label1

المرتب

Label2

(General)

(Declarations)

```
Private Type Employee
    sFName As String
    sLName As String
    fPayHour As Single
    fCut As Single
    fSalary As Single
    nHours As Integer
    bCase As Boolean
End Type
```

```
Private Sub CmdCalcPay_Click()  
Dim Emp As Employee  
'receive data from user and assign valuse for variables  
Emp.sFName = TxtFName.Text  
Emp.sLName = TxtLName.Text  
Emp.fPayHour = Val(TxtPayHour.Text)  
Emp.nHours = Val(TxtNumHours.Text)  
If OptCut.Value = True Then  
    Emp.bCase = True  
Else  
    Emp.bCase = False  
End If  
'calculate cuts  
If Emp.bCase = False Then  
    Emp.fCut = 0  
ElseIf Emp.fPayHour <= 100 And Emp.fPayHour >= 90 Then  
    Emp.fCut = 0.15 * Emp.fPayHour * Emp.nHours  
ElseIf Emp.fPayHour < 90 And Emp.fPayHour >= 75 Then  
    Emp.fCut = 0.1 * Emp.fPayHour * Emp.nHours  
ElseIf Emp.fPayHour < 75 And Emp.fPayHour >= 50 Then  
    Emp.fCut = 0.08 * Emp.fPayHour * Emp.nHours  
ElseIf Emp.fPayHour < 50 Then  
    Emp.fCut = 0.05 * Emp.fPayHour * Emp.nHours  
Else  
    MsgBox ("You entered invalid Hour rate")  
End If
```


حل مثال 2: الكود:

```
'calculate net payment or salary
Emp.fSalary = Emp.fPayHour * Emp.nHours - Emp.fCut
'show employee's data on Form2
Form2.LblName.Caption = Emp.sFName & " " & Emp.sLName
Form2.LblSalary.Caption = Emp.fSalary
Form2.Show
End Sub
```

حل مثال 2: وضع التنفيذ 1:

Form1

الاسم الأول: أمير

الاسم الأخير: يوسف

أجر الساعة: 65.5

عدد الساعات: 18

الخصومات:

☒ يوجد

☐ لا يوجد

Show Payroll

Form2

اسم الموظف	أمير يوسف
المرتب	1084.68

حل مثال 2: وضع التنفيذ 2:

Form1

الاسم الأول عمرو

الاسم الأخير أنور

أجر الساعة 37.80

عدد الساعات 20

الخصومات

☐ يوجد

☒ لا يوجد

Show Payroll

Form2

اسم الموظف	عمرو أنور
المرتب	756

■ بدلا من كتابة `Emp.fPayHour * Emp.nHours` عدة مرات قم بعمل متغير خاص داخل `CmdCalcPay_Click()` يحمل هذه القيمة.

■ تعديل في حل المثال السابق:

قم بعمل متغيرين عموميين `Public` في حيز الاعلان العام للنموذج الأول، المتغير الأول يحمل اسم الموظف كاملا بينما يشير المتغير الآخر الى صافي المرتب. يتم ضبط قيمتي هذين المتغيرين في `CmdCalcPay_Click()` بينما يمكن استدعاءهما في اجراء حدث تحميل النموذج الثاني. وبالتالي يمكن الاستغناء عن هذا الجزء من الكود:

```
Form2.LblName.Caption = Emp.sFName & " " & Emp.sLName  
Form2.LblSalary.Caption = Emp.fSalary
```

■ صمم برنامجا تعليميا يستقبل عددين صحيحين و يتم اختيار أحد العمليات الجمع أو الطرح أو الضرب أو القسمة من صندوق موحد و كذلك يستقبل النتيجة من المستخدم، وإذا كانت اجابة المستخدم صحيحة يتم عرض صورة مناسبة تعبر عن الاجابة الصحيحة والا يتم عرض علامة خطأ و كذلك صندوق رسائل يعرض الاجابة الصحيحة. ضمن البرنامج زر للخروج مع سؤال للتأكد من الرغبة في الخروج.

■ استخدم متحكمات صور بحيث تظهر للمستخدم.

■ لا تظهر كلتي الصورتين عند بدء البر

- كيفية الإعلان عن المتغيرات في الـ VB ؟
- استخدام إيعاز If & if – else (اتخذ القرار).