

بسم الله الرحمن الرحيم

الدرس الاول:

أخوتي هذه الدروس عبارة عن أساسيات وقواعد الفيجوال بيسك حيث أنني لن أتعمق كثيرا لضيق الوقت .

بناء على طلب وأستعجال الاخوة انا مضطر الى أختصار الكثير من الدروس لاستعجالهم ، كنت أتمنى أن أكتب الدروس على أكمل وجه .

وبناء على طلب بعض الاخوة دروس حول قواعد البيانات سأمدهم أنشاء الله في الايام القادمة بالدروس عن قواعد البيانات بأنواعها . أيضا كان بودي لو أنني وضعت هذه الدروس ضمن هذا الدرس .

لكن تعبت يداي بجد وما كان هناك الوقت الكافي وشكرا لكم .

أخواني هذا الكتاب لقد كتبتة باللغة العربية الفصحى ، حتى يكون مختصر أكثر ، يجب عليك فهم أستيعاب الكلام وفهمه على أكمل وجه وأضمن لك النجاح .

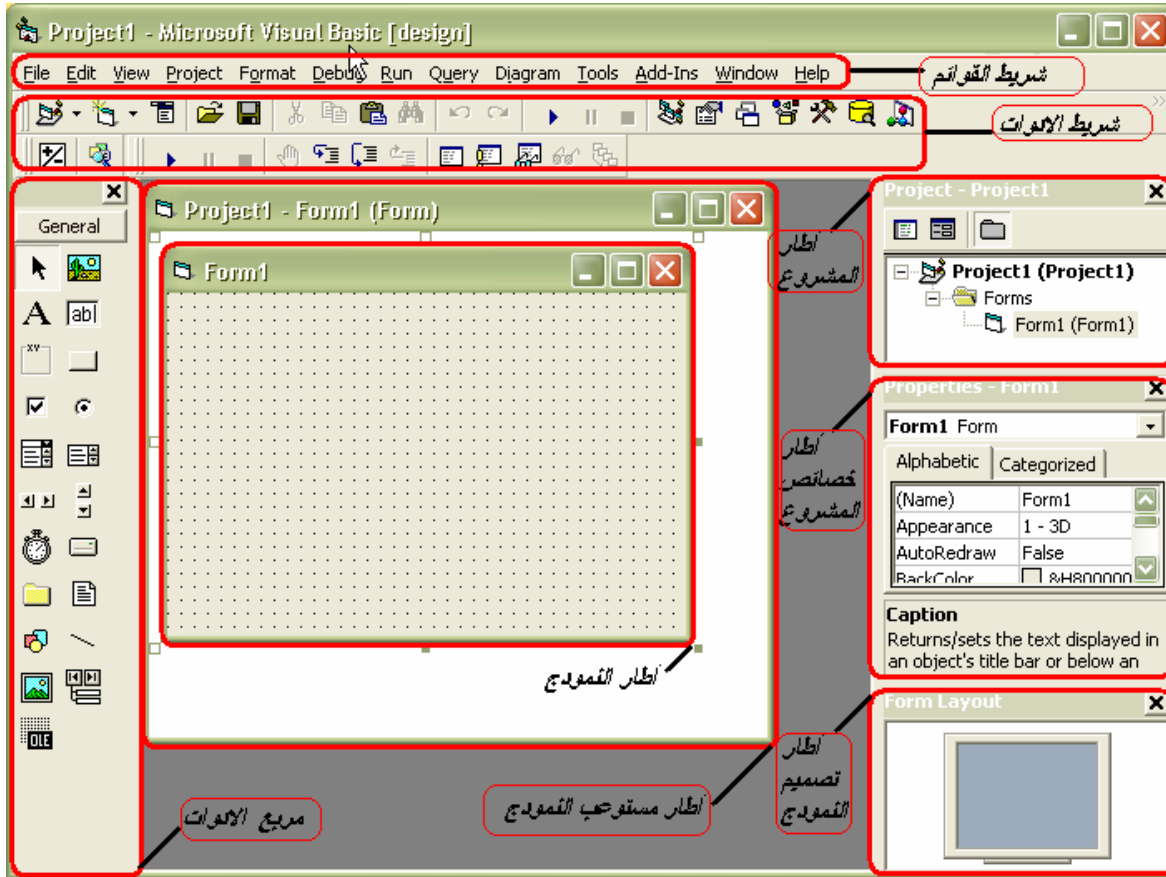
نبدأ ولا نكثر الكلام خير الكلام ما قل ودل .

الفهرس

- ١ . شرح واجهة البرنامج فيجوال بيسك .
- ٢ . شرح واجهة New Project .
- ٣ . شرح واجهة عناصر مربع الادوات .
- ٤ . شرح واجهة أطار الخصائص الادوات .
- ٥ . شرح واجهة أطار المشروع .
- ٦ . أستعمال بينيات القرار .
  - أختبار عدد الشروط في بنية القرار If...Then .
  - التحقق من هوية المستخدم بأستعمال If...Then .
  - أستعمال العوامل المنطقية في التعابير الشرطية .
  - بنية القرار Select Case .
  - أستعمال عوامل المقارنة مع البنية Select Case .
- ٧ . أستعمال حلقات
  - كتابة حلقات For...Next .
  - عرض معلومات بأستعمال حلقة For...Next .
  - تغيير خاصية في الحلقة For...Next .
  - أنشاء حلقات For...Next معقدة .
  - الجمل For...Next المركبة .
  - الجملة Exit For .
  - كتابة حلقات Do .
  - تجنب الحلقة اللامتناهية .
  - تحويل درجات الحرارة بأستعمال حلقة Do .
  - أستعمال الكلمة الدلالية Until في حلقة Do .
- ٨ . المتغيرات والعوامل في فيجوال بيسك
  - هيكلية جمل الفيجوال بيسك البرمجية .
  - أستعمال المتغيرات لتخزين المعلومات .
  - حجز مساحة للمتغيرات بأستعمال الجملة Dim .
  - تصريح المتغير من دون أستعمال الجملة Dim .

- استعمال متغيرات لتخزين الادخال .
- عرض رسالة بأستعمال الدالة MsgBox .
- العمل مع أنواع معينة مع البيانات .
- الثوابت ( المتغيرات التي لا تتبدل ) .
- ٩ . أستعمال عوامل فيجوال بيسك .
- دالات فيجوال بيسك الرياضية .
- أستعمال العوامل المتقدمة .
- أولوية العوامل .
- خطوة متقدمة أستعمال الاولوية للعوامل .

١٠ . أدعولي



لمعرفة وظيفة كل زر في شريط الاموات ، ضع مؤشر الماوس فوقه ليرفه

برنامج القيجوال بيسك ٦,٠ Visual Basic 6.0

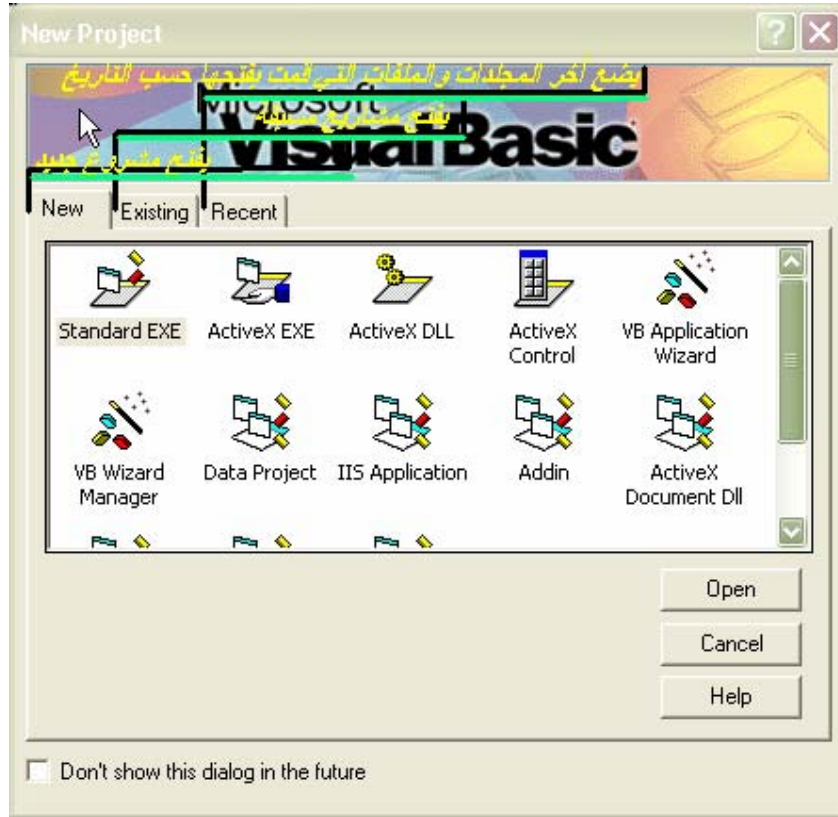
### شرح برنامج فيجوال بيسك

تحتوي بيئة فيجوال بيسك على أدوات برمجة لمساعدتك على بناء برامج فيجوال بيسك . يزود شريط القوائم وصولاً إلى معظم الأوامر التي تتحكم ببيئة البرمجة فيجوال بيسك . تعمل القوائم والأوامر في فيجوال بيسك مثلما تفعل في كل البرامج المتوافقة مع ويندوز ، وبأستطاعتك الوصول إليها بواسطة الماوس أو لوحة المفاتيح . تحت شريط القوائم يوجد شريط الادوات ، وهو عبارة عن مجموعة من الازرار التي تشكل اختصارات لتنفيذ الاوامر والتحكم ببيئة البرمجة فيجوال بيسك . يجب أن تكون معتاداً على شريط الادوات إذا أستعملت مايكروسوفت وورد أو مايكروسوفت أكسل من قبل .

لتنشيط أحد أزرار شريط الأدوات ، انقره بواسطة الماوس .  
بالإضافة الى ذلك يوجد مربع الأدوات و أطار مستوعب المشروع و أطار النموذج و أطار المشروع و أطار الخصائص و أطار تصميم النموذج .  
سنتعلم كيفية أستعمال هذه الميزات لتخصيص بيئة البرمجة بهذا الشكل لاحقا ولا يتم الاعتياد على هذه الأدوات إلا من خلال أستعمالنا لها بأذن الله تعالى

### شرح واجهة New Project

مربع حوار مشروع جديد هو عبارة عن نافذة تظهر للمبرمج عند البدء بالدخول الى بيئة البرمجة فيجوال بيسك .  
ويتم منها أختيار المشروع القياسي الذي يريد البدء فيه ، كما ترى هناك العديد منها كل مشروع له عمله ، والمشروع الافتراضي دائما هو Standard EXE ألا اذا المبرمج أختار مشروع آخر ليبدء به العمل .  
طبعا يوجد ثلاث خيارات في مربع الحوار New Project الاول وهو (New) يترك لك الخيار لتختار المشروع الذي يناسبك مشروع جديد .  
Existing الخيار الثاني يفتح لك نافذة البحث داخل جهازك لفتح مشروع قديم محفوظ على جهازك .  
Recent أما الخيار الثالث يضع آخر المجلدات والمشاريع التي قمت بفتحها حسب التاريخ .  
حسنا كما ترى الجزء الوحيد الذي سيحتاج الى الشرح هو الخيار الاول (New) مشروع جديد .



مربع حوار New Project

أول ما يظهر للمبرمج عند دخوله هو مربع حوار أسمه (مشروع جديد New Project) . ليترك المجال أمام المبرمج أختيار نوع المشروع الذي سيعمل عليه .

وكما نرى فوق يوجد ١٣ نموذج وكل نموذج له مميزات خاصة به سأشرح بأختصار ما عمل كل نوع من النماذج السابقة .

### الشرح بأختصار

المشروع	الشرح
Standard EXE	وهو يقوم بإنشاء برنامج تنفيذي بعد طلبك لذلك من ( Make EXE ) في قائمة ( File )
Activex EXE	وهو أيضاً يقوم بإنشاء ملفاً له الامتداد ( DLL ) وهو ملف ذو برامج دعيه مساعدة و هو لا يعمل بنفسه بل مع .exe
Activex EXE	وهو ينشئ ملفات لها امتداد EXE تعمل مع برامج أخرى لكي يعمل في شكل ( OLE )
Activex control	وهو يساعد على إنشاء ملف من نوع ( ocx ) و هي مهمة وتساعد على إحتواء واجهة مستخدم أو برامج فرعية.
Activex Document DLL	وهو يساعد على إنشاء ملف ذو امتداد ( DLL ) يقوم بتشغيل برامج على موقع في الانترنت
Activex Document exe	يقوم بإنشاء ملفات تظهر على الانترنت.
Add In	وهو خاص بالفيجوال بيسك حيث يمكنك إضافة واجهة مع ما ذكر سابقاً خاصة بك.
VB Application Wizard	وهو معالج تلقائي في هذه اللغة يساعدك على إنشاء نوافذ عدة دون كتابة سطر واحد من النوع
SS Application	نقصد بـ ( ISS ) إختصار لكلمة ( Internet Information Server ) و يقوم بالتحضير لتصميم ( web class ) وهو ما يسميه المستخدم صفحات ( HTML )
Data project	تضع مشروعاً لقواعد بيانات وهيئة في بيئة متكاملة من أدوات الربط وصنع التقارير دون الحاجة لفتح برامج أخرى .
Vb wizard manager	و تبدو مشابهة لمعالجات ( Microsoft ) يمكنك إستخدامه
DHTML Application	تسهل عليك صنع صفحات ( HTML ) ديناميكية للإنترنت دون تعلم لغة .Java
Vb Enterprise Edition controls	وهي تساعد على وضع جميع الأدوات المستخدمة بكثرة لقواعد البيانات و الاتصال ومن ثم تطبيقها وهي للمحترفين

أنقر على الزر Open للموافقة على النوع الافتراضي وهو برنامج فيجوال بيسك ٦,٠ ( ٣٢ بت قياسي ) Standard EXE.

يتم فتح مشروع جديد في بيئة البرمجة فيجوال بيسك .  
والآن لقد شرحنا واجهة المستخدم بأختصار حتى لا يطول علينا وعليكم الوقت ، ولو أنني شرحت كل نوع لوحده لطلال بنا المطال ، ما يهمنا هنا هو أن نكون على دراية بالانواع السابقة .  
الآن سننتقل الى الدرس التالي ألا وهو شرح عناصر مربع الادوات

### عناصر مربع الأدوات

للتعرف على بعض العناصر الموجودة في مربع الأدوات بداية نقول أن هذه العناصر تسمى أدوات Active X وهي أدوات جاهزة صنعتها لك مايكروسوفت لتساعدك في إنشاء البرامج . أنها مثل ActiveX Control هذا المشروع يخلق أداة أمتداده OCX وعند تحريره فأن فيجوال بيسك يتيح لك حفظها كأمتداد OCX وليس EXE وتستطيع تحميلها بعد تحريرها الى بيئة فيجوال ، وعند تحميلها تظهر على مربع الأدوات مثلها مثل أي أداة أو عنصر على مربع الأدوات ، ولكن طبعا هذا المشروع نحن من قام بعمله ويمكن أن يحمل خصائص الأدوات القياسية مثل أي أداة على مربع الأدوات ، ويمكن أن لا يحمل هذه الخصائص ، هذا يرجع الى مبرمجها .

ما يهنا هنا هو أن مربع الأدوات هو ليساعدك على تطوير برنامجك ، هذه الأدوات أنك تراها من جهة مسطحة ، فأنا لو تعمقنا بها لنجد أن كل أداة مصنوعة من أكواد كبيرة وكثيرة ، شركة المايكروسوفت قد سهلت الامر علينا بجعلها سهلة المنال ، وبمجرد النقر على أي أداة على مربع الأدوات ستظهر في النموذج المشروع دعونا الآن نبدأ بالشرح :-



شرح مختصر :-

الشرح	الأداة
وهي تعيد مؤشر الماوس الى السهم الطبيعي اذا كان المؤشر على احدى الأدوات أو الاشكال الاخرى غير الوجه الطبيعي .	Pointer
وهي أداة تتيح لك إضافة صورة الى البرنامج وأدمج الصورة الى البرنامج .	PictureBox
وهي أداة تتيح لك إضافة عنوان ما الى البرنامج ، أو نص في أي مكان من الفورم ، وقد تستخدم ال Label في منح المستخدم بعض المساعدة للدلالة الى شيء ما .	Label

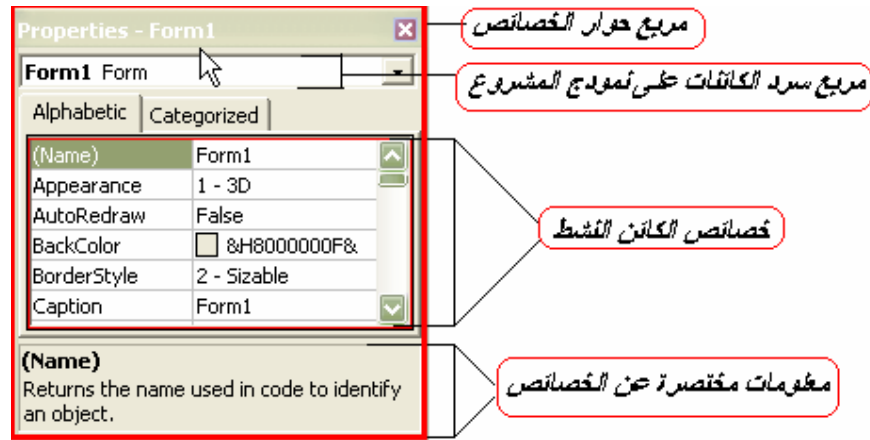
وهي أداة تتيح لك أن تترك للمستخدم فرصة إدخال بيانات مثل اسمه مثلا أو مسار البرنامج أو شيء آخر .	Text Box
وهي أداة تتيح لك صنع Frame إطار وتضمن بعض الأدوات بداخل هذا الإطار .	Frame
وهي أداة زر امر حيث تتيح لك أن تجعل للمستخدم زر ليضغط عليه عند تنفيذ أمر معين .	Command Button
وهي أداة مربع اختيار ، لتتيح للمستخدم اختيار أكثر من خيار .	Check Box
وهي أداة زر راديو ، لتتيح للمستخدم من اختيار خيار واحد من عدة خيارات .	Option Button
وهي أداة تضيف إليها قائمة منسدلة مثلا ليختار منها المستخدم إحدى الخيارات المتاحة أو القيم المتاحة .	Combo Box
وهي قائمة تشبه الى حد ما ال Combo Box لكنها ليست قائمة منسدلة مثل الاداة السابق ذكرها .	List Box
وهي أداة شريط الانزلاق العرضي ، وتستخدم في أنزلاق الصور والكائنات والاكبر من اللازم عرضيا .	HScrolBar
وهي أداة تشبه السابقة ولكن شريط أنزلاق طولي وليس عرضي .	VScrollBar
وهي أداة المؤقت وتوظيفها أنها تقوم بأداء عمل معين أو عدة أعمال معينة بصفة دورية كلما مر زمن معين تحدده .	Timer
وهي أداة عبارة عن List Box فيها أقسام القرص الصلب وقسمي القرص المرن والي دي .	Drive List Box
وهي عبارة عن أداة List Box وتقوم بعرض المجلدات Folders في مسار معين تحده أنت .	Dir List Box
وهي عبارة عن أداة List Box تقوم بعرض الملفات Files في مسار معين .	File List Box
وهي عبارة عن أداة رسم شكل هندسي .	Shape
وهي أداة رسم خط على الفورم .	Line
وهي أداة إضافة صورة وتختلف بعض الاختلافات عن الاداة Picture Box في الخصائص بينهما .	Image
وهي أداة تستخدم في ربط البرنامج بقاعدة بيانات خارجيه .	Data
وهي أداة ربط وتضمن ملفات وبرامج خارجيه ضمن برنامجك .	OLE

لا تخف أخي المتدرب سنأتي الى ذكر كل أداة على حدة ، حتى تعمل على أدارتها والتكيف مع هذه الادوات .  
وهذا سيكون من خلال أمثله عامة وليست خاصة بأي أداة حتى نختصر الشرح المطول وحتى أرتاح من الكتابة  
الآن لا بد من شرح بعض خصائص الادوات .  
الى الدرس التالي بأذن الله تعالى :-

### الدرس الثالث

#### شرح إطار الخصائص الأدوات

أطار الخصائص هو عبارة عن خصائص عناصر الادوات التي تعمل عليها حاليا ، مثلا لو أنك فتحت مشروع جديد فأن أطار الخصائص فوراً يظهر لك خصائص النافذة أو المشروع الجديد .  
 أن الفورم هو عبارة عن أداة مثل الادوات التي تستعملها مثل Command Button أو Label أو أي أداة موجودة داخل بيئة العمل أو بيئة فيجوال بيسك .  
 وتختلف خصائص الادوات من أداة الى أخرى ، فمثلا خصائص الفورم تختلف عن خصائص PictureBox ولو أن هناك بعض الخصائص المتشابهة التي تحملها كل الادوات تقريبا .  
 مثلا أن الفورم له خاصية تدعى Name وطبعا كل الادوات تحمل هذه الخاصية ، وبغض النظر عن عمل هذه الاداة التي تقوم به .  
 دعونا الآن ننظر الى بعض الخصائص وشرح مختصر عنها :-



أطار مربع خصائص الاداة

الخاصية	شرحها
Name	تحديد أسم فريد للأداة
BackColor	لون الخلفية
BorderStyle	سمك أطار النافذة
Caption	عنوان أداة
Text	عنوان أداة
ControlBox	ما يظهر في اعلى كل نافذة في اقصى اليمين للتحكم بالنافذة
Enabled	لتمكين الاداة و الغاء تمكينها
Font	الخط
ForeColor	اللون الامامي
Height	الارتفاع
Icon	رمز
Left	موقع الاداة من اليسار
Mdichild	تستخدم في النوافذ المتعددة النوافذ
MousePointer	مؤشر الفأرة
Picture	الصورة
RightToLeft	مفيدة جدا في البرامج العربية لتحويل القوائم من اليسار الى اليمين
Top	موقع الاداة من فوق
Visible	أخفاء الاداة وأظهارها
Width	عرض الاداة

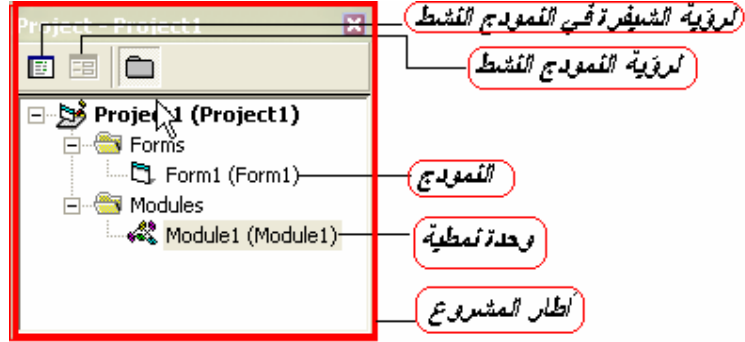
حالة النافذة من حيث الحجم	WindowState
جعل الاداة شفافة	BackStyle
حذف الاطار المحيط بالاداة	BorderStyle
أسم قاعدة البيانات (لربط)	DataBaseName
seek اذا استخدمنا البحث في قاعدة البيانات بالامر table يجب ان توضع في حالة	RecordSet
تحديد الجدول من قاعدة البيانات	RecordSource
ربط أداة النص بحقل معين من قاعدة البيانات	DataField
ربط أداة النص بجدول معين من قاعدة البيانات	DataSource
القيمة	Value
الحد الاعلى	Max
الحد الادنى	Min
التغيير الصغير	SmallChange
التغيير الكبير	LargeChange
الفترة الزمنية المراد تنفيذ حدث المؤقت بعد مرورها	Interval
التعليق الذي يظهر فوق الاداة عند ابقاء الفارة عليا لمدة قصيرة	ToolTipText
على زر معين tabتوقيف التنقل بزر	TabStop
ترتيب التنقل بين الادوات	TabIndex

الآن سننتقل الى الدرس التالي اطار المشروع :

### أطار المشروع

يتألف برنامج المشروع من عدة ملفات مجموعة سويًا ، أو **مصرفة** ، عندما يكتمل البرنامج . لمساعدتك على التبديل بين المكونات خلال العمل على مشروع ، أضاف مصمموا فيجوال بيسك **أطار المشروع** الى بيئة البرمجة ( يسمى بعض الأشخاص هذا الاطار **مستكشف المشروع** ) يسرد اطار المشروع كل الملفات المستعملة في عملية البرمجة ويؤمن وصولا اليها من خلال زر **ر**ين **ه**ما **ر**ؤية **الكائنات** و**رؤية الشيفرة** . عندما تضيف ملفات فردية الى المشروع وتحفظها فيه أو تحذفها منه بأستعمال أوامر القائمتين File و Project ، ستبدوا هذه التغييرات في اطار المشروع . يسمى ملف المشروع ( الذي يحتوي على لائحة بكل الملفات المساعدة في مشروع برمجة ) ملف **مشروع فيجوال بيسك** وله ملحق (vbp) . في إصداري فيجوال بيسك ٥ و ٦ ، يمكن تحميل أكثر من ملف مشروع واحد في اطار المشروع بنفس الوقت ، ويمكنك التنقل بينها عبر نقر أسم المشروع . تحت اسم المشروع ، يعرض اطار المشروع مكونات كل مشروع في بنية شجرية مشابهة لتصميم مستكشف ويندوز . يمكنك توسيع أو طي فروعها ، بما في ذلك النماذج والوحدات النمطية وغيرها من الفئات ، عبر نقر العلامات + و- بجانب المجلدات ، أنظر الشكل التالي :





ويقوم بعرض جميع المشاريع و النماذج للمشروع

الحالي

الآن وتقريباً قد أستعرضنا واجهة البرنامج وأدواته التي دائماً ما سنتعامل معها .  
والآن سننتقل إلى الدرس التالي بنيات القرار :-

## بنيات القرار

تستطيع التعبيرات الشرطية أن تتحكم بالترتيب الذي يتم فيه تنفيذ الجمل عند أستعمالها ضمن كتلة من الجمل تعرف ببنية القرار . تتيح لك بنية القرار If...Then تقييم شرط في البرنامج وأخذ خط عمل معين وفقاً للنتيجة . تكتب بنية القرار If...Then في أبسط أشكالها على سطر واحد :

### If (condition) Then (statement)

حيث أن *condition* هو التعبير الشرطي و *statement* هي جملة برمجية فيجوال بيسك صالحة .  
مثلاً :

If Score >= 20 Then Label1.caption = "You win!"

الجملة السابقة هي جملة بنية قرار If...Then تستعمل التعبير الشرطي

Score >= 20

لتحديد ما اذا كان على البرنامج ان يضبط الخاصية Caption (أنها خاصة التسمية) للكائن Label1 عند "You win!" أم لا . إذا أحتوى المتغير Score على قيمة أكبر أو تساوي ٢٠ ، يضبط فيجوال بيسك الخاصية Caption وألا يتخطى جملة الضبط وينفذ السطر التالي من الاجراء الحداثي . ينتج عن هذا النوع من المقارنات دائماً قيمة صح أو خطأ . لا ينتج أبداً عن التعبير الشرطي قيمة "ربما" .

### أختبار عدد من الشروط في بنية القرار If...Then

يدعم فيجوال بيسك أيضاً بنية قرار If...Then تتيح لك شمل عدة تعابير شرطية . يمكن أن تبلغ كتلة الجمل هذه عدة أسطر وتحتوي على الكلمات الدلالية ElseIf و Else و End . If القاعدة العامة :-

If condition Then

Statement executed if condition1 is True

ElseIf condition2 Then

```
Statement executed if condition2 is True
[Additional ElseIf clauses and statements can be placed here]
Else
Statements executed if none of the conditions is True
End If
```

في هذه البنية ، يتم تقييم *condition1* أولا . إذا كان التعبير الشرطي صح ، يتم تنفيذ كتلة الجمل التي تليه ، جملة تلو الاخرى (يمكنك وضع جملة برمجية واحدة أو أكثر). وإذا لم يكن الشرط الاول صح ، يتم تقييم التعبير الشرطي الثاني (*condition2*) .  
إذا كان الشرط الثاني صح ، يتم تنفيذ الكتلة الثانية من الجمل .  
(يمكنك إضافة المزيد من الشروط والجمل ElseIf وإذا كانت هناك شروط أكثر يجب تقييمها).  
أخيرا إذا لم يكن أي من التعابير الشرطية صح ، يتم تنفيذ الجمل التي تلي الكلمة الدلالية Else .  
أخيرا ، يتم إنهاء البنية بأكملها بالكلمات الدلالية End If .  
دعونا ننظر الى الشيفرة التالية والتي تستعمل بنية القرار If...Then المتعددة الاسطر لتحديد مقدار الضريبة المتوجبة في ضريبة تصاعدية أفتراضية :

```
If AdjustedIncome <= 24650 Then
TaxDue = AdjustedIncome * 0.15
ElseIf AdjustedIncome <= 59750 Then
TaxDue = 3697 + ((AdjustedIncome – 24650) * 0.28)
ElseIf AdjustedIncome <= 124650 Then
TaxDue = 13525 + ((AdjustedIncome – 59750) * 0.31)
ElseIf AdjustedIncome <= 271050 Then
TaxDue = 33644 + ((AdjustedIncome – 124650) * 0.36)
Else
TaxDue = 86348 + ((AdjustedIncome – 271050) * 0.396)
End If
```

ملاحظه هامه:-

عندما تستعمل أكثر من تعبير شرطي واحد عليك الانتباه الى ترتيب تلك التعابير .  
دعونا الآن نشرح تلك الشيفرة بعض من التفصيل .  
تختبر بنية القرار هذه المتغير AdjustedIncome عند مستوى الدخل الاول ثم عند المستويات التالية الى أن يتم تقييم أحد التعابير الشرطية صح فتحسب قيمة ضريبة الدخل المتوجبة على الفرد . مع بعض التعديلات البسيطة ، يمكن أستعمالها لاحتساب الضريبة المتوجبة على كل فرد في نظام ضرائب تصاعدية .  
شرط أن تكون الضرائب كاملة ومحدثة وأن تكون القيمة في المتغير AdjustedIncome صحيحة .

إذا تغيرت معدلات الضرائب يكون من السهل تحديث التعابير الشرطية .  
سنستعمل في التمرين التالي بنية قرار If...Then لنتحقق من هوية المستخدم أثناء تسجيل دخولهم الى البرنامج . قد تستعمل منطقا مشابها إذا كنت تكتب تطبيقا سيتم أستعماله في شبكة كومبيوترات .

التحقق من هوية المستخدمين بأستعمال If...Then

(١)أفتح مشروع جديد

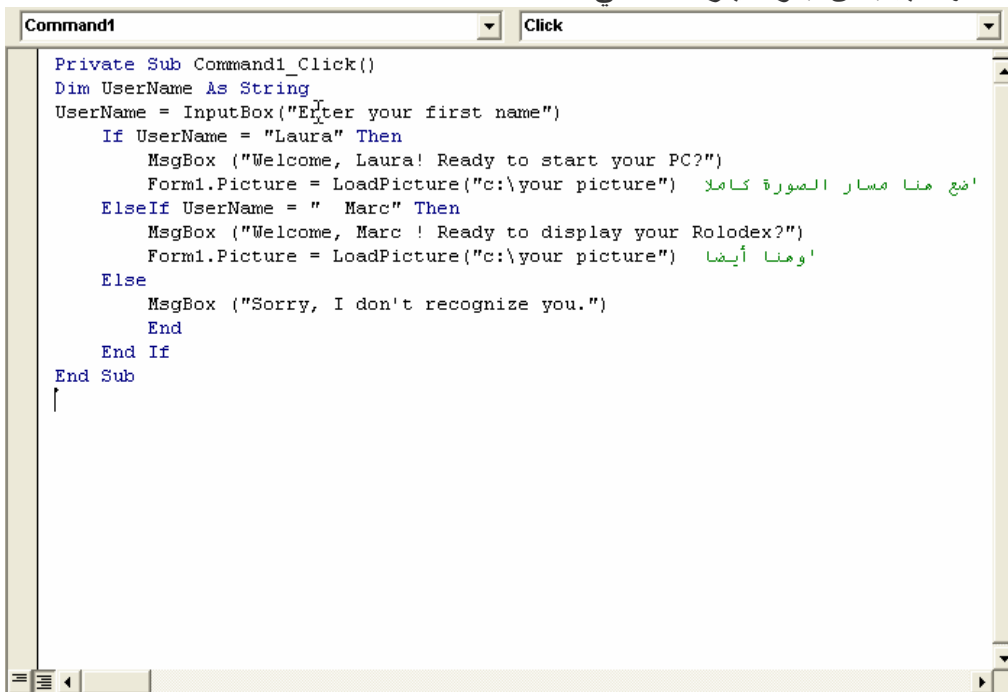
(٢)أضف زر أمر Command1

(٣)أضبط الخاصية Caption عند Log in

٤) أنقر الزر Log in نقرأ مزدوجا سيظهر الاجراء الحداثي Command1\_Click في محرر الكود أو أطار الشيفرة  
٥) أكتب الجمل البرمجية التالية في الاجراء :

```
Dim UserName as String
UserName = InputBox("Enter your first name")
If UserName = "Laura" Then
    MsgBox ("Welcome, Laura! Ready to start your PC?")
    Form1.Picture = LoadPicture ("c:\your picture") 'ضع هنا مسار الصورة ' كاملا
ElseIf UserName = " Marc" Then
    MsgBox ("Welcome, Marc ! Ready to display your Rolodex?")
    Form1.Picture = LoadPicture ("c:\your picture") 'وهنا أيضا'
Else
    MsgBox ("Sorry, I don't recognize you.")
End
End If
```

بعد الانتهاء يجب أن تبدوا شيفرتك كالتالي :



```
Private Sub Command1_Click()
Dim UserName As String
UserName = InputBox("Enter your first name")
If UserName = "Laura" Then
    MsgBox ("Welcome, Laura! Ready to start your PC?")
    Form1.Picture = LoadPicture("c:\your picture") 'ضع هنا مسار الصورة كاملا'
ElseIf UserName = " Marc" Then
    MsgBox ("Welcome, Marc ! Ready to display your Rolodex?")
    Form1.Picture = LoadPicture("c:\your picture") 'وهنا أيضا'
Else
    MsgBox ("Sorry, I don't recognize you.")
End
End If
End Sub
```

محرر الكود في فيجوال بيسك

الآن دعنا نشغل البرنامج ، أضغط على زر تشغيل



أنقر على الزر Log in تعرض الدالة InputBox المكتوبة في الاجراء الحداثي Command1\_Click مربع حوار يطلب منك أن تكتب أسمك الاول . أكتب Laura ثم أضغط . Enter

تقارن بنية القرار If...Then الاسم الذي كتبته بالنص "Laura" في التعبير الشرطي الاول من الكود السابق . أذا كتبت Luara ، سيقوم التعبير الى صح وتعرض الجملة If...Then رسالة ترحيب بأستعمال الدالة MsgBox .  
أنقر على OK في مربع الرسالة.

ينغلق مربع الرسالة ويتم تحميل ملف الصورة التي وضعت مسارها في الكود .  
لقد حملنا الفورم صورة من خلال الكود ، من خصائص الفورم يوجد خاصية تدعى Picture هذه الخاصية تحمل على الفورم صورة أنت تحددها تستطيع تحميل من أطار خصائص الكود مباشرة دون كتابة أي كود ، هذا الاطار على يمين المشروع .

### أستعمال العوامل المنطقية في التعابير الشرطية .

يتيح لك فيجوال بيسك أختبار أكثر من تعبير شرطي واحد في القسمين If...Then و ElseIf أذا أردت شمل أكثر من معيار أنتقاء واحد في بنية القرار . يتم ربط الشروط الاضافية سوية بأستعمال واحد أو أكثر من العوامل المنطقية التالية :

العامل المنطقي	معناه
And	أذا كان التعبيران الشرطيان صح ، تكون النتيجة صح.
Or	أذا كان أحد التعبيرين الشرطيين صح ، تكون النتيجة صح .
Not	أذا كان التعبير الشرطي خطأ ، تكون النتيجة صح. وأذا كان التعبير الشرطي صح ، تكون النتيجة خطأ
Xor	صح . وأذا كان كلاهما صح أو خطأ ، فالنتيجة خطأ.

تلميح :

عندما يقيم برنامجك تعبيراً معقداً يتضمن أنواعاً مختلفة من العوامل ، فإنه يقيم العوامل الرياضية أولاً ثم عوامل المقارنة ثم العوامل المنطقية.

يسرد الجدول التالي بعض الامثلة عن عوامل منطقية خلال عملها . يفترض في التعابير أن المتغير Vehicle يحتوي على القيمة "Bike" والمتغير "price" على القيمة 200 .

التعبير المنطقي	النتيجة
Vehicle = "Bike" And Price < 300	صح ( كلا التعبيرين صح )
Vehicle = "Car" Or Price < 500	صح ( شرط واحد صح )
Not Price < 100	صح ( الشرط خطأ )
Vehicle = "Bike" Xor Price < 300	خطأ ( كلا الشرطين صح )

ستعدل في التمرين التالي على البرنامج السابق  
ضع السطر التالي بين الجملة InputBox والجملة If...Then في الاجراء (بين السطرين الاول والثاني):

Pass = InputBox ("Enter your password.")

عدل هذا السطر في البرنامج السابق :

If UserName = "Laura" And Pass = "May17" Then

تتضمن الجملة الآن العامل المنطقي And الذي يتحقق من أسم المستخدم ومن كلمة المرور قبل قبول Laura في البرنامج .

عدل أيضا الجملة ElseIf الى الشكل التالي :

ElseIf UserName = Marc" And Pass = "trek" Then

يضيف العامل And أختبارا لكلمة المرور "trek" التي تخص المستخدم Marc .

جرب البرنامج الآن بعد التعديل .

### بنية القرار Select Case .

يتيح لك فيجوال بيسك التحكم بتنفيذ الجمل في برامجك بأستعمال بنيات القرار Select Case . تشبه البنية Select Case البنية If...Then...ElseIf لكنها أكثر فعالية عندما يعتمد التفرع على متغير رئيسي واحد ، أو حالة اختبار . يمكنك أيضا أستعمال البنية Select Case لتسهيل قراءة البرنامج ولجعله أكثر فعالية . تركيبها النحوى كالتالى :

Select Case Variable

Case Value1

Program statements executed if value1 matches variable

Case Value2

Program statements executed if value2 matches variable

Case Value3

Program statements executed if value3 matches variable

.  
. .  
. .

End Select

دعونا نشرح بعض من التفصيل

تبدأ البنية Select Case بالكلمتين الداليتين Select Case وتنتهي بالكلمتين End Select . عليك أستبدال Variable بالمتغير أو الخاصية أو التعبير الذي سيكون القيمة الدليلية أو حالة الاختبار في البنية . وأستبدل Value1 و Value2 و Value3 بالأرقام أو بالسلاسل أو بالقيم الأخرى التي لها صلة بحالة الاختبار التي تتم دراستها . اذا طبقت إحدى القيم المتغير ، يتم تنفيذ الجمل الموجودة تحت تعبيرها Case ويتابع فيجوال بيسك تنفيذ شيفرة البرنامج بعد Select Case . يمكنك أدرج أي عدد تريده من تعابير Case كما يمكنك إضافة أكثر من قيمة واحدة في كل تعبير Case .أذا دكرت عدة قيم بعد حالة ما ، أفصل بينها بفواصل . أنظر الى المثال التالي :

Dim Age As Integer

Age = 21

Select Case Age

Case 16

Label1.Caption = "You can drive now!"

Case 18

Label1.Caption = "You can vote now!"

Case 21

Label1.Caption = "You can drink wine with your meals."

Case 65

Label1.Caption = "Time to retire and have fun!"

End Select

تدعم بنية Select Case أيضا استعمال تعبير Case Else الذي يمكنك أستعماله لأظهار رسالة في حال عدم مطابقة أي من الحالات السابقة .  
أليك أستعمالها في مثال العمر :

Dim Age As Integer

Age = 21

Select Case Age

Case 16

Label1.Caption = "You can drive now!"

Case 18

Label1.Caption = "You can vote now!"

Case 21

Label1.Caption = "You can drink wine with your meals."

Case 65

Label1.Caption = "Time to retire and have fun!"

Case Else

Label1.Caption = "You're a great age! Enjoy it!"

End Select

شرح مبسط :

لقد صرحنا متغير اسمه Age وأعطيناها نوع رقم صحيح Integer بعد ذلك أعطينا المتغير رقم 21 .  
تختبر بنية القرار المتغير Age ، وتنظر الى ما يحمله المتغير هنا في مثالنا هذا المتغير يحمل الرقم 21 .  
يختبر الاول أذا كان يحمل الرقم 16 ثم الثاني 18 الثالث يحمل الرقم 21 ، أذا هنا بنية القرار سوف تعرض في أداة التسمية الجملة التالية:  
(You can drink wine with your meals.)  
وتنتقل الى آخر البنية الأ وهي End Select ، طبعا لاحظنا وجود الجملة Case Else ، هذا يعني أذا لم تطابق أي من القيم في بنية القرار Select Case ستنقل الى الجملة Case Else لأنه لا يوجد ما يتطابق مع القيم التي وضعناها في البنية هذا أختبار جيد ، في حال حصل شيء ليس في الحسابان فتنفذ الجملة Case Else .

### أستعمال عوامل المقارنة مع البنية Select Case .

يتيح لك فيجوال بيسك أستعمال عوامل المقارنة لأدراج نطاق من قيم الأختبار في البنية Select Case .

عوامل المقارنة التي يمكن أستعمالها هي = و < و > و <= و >= .  
لأستعمال عوامل المقارنة عليك كتابة الكلمة الدليلية Is أو To في التعبير لتعريف المقارنة التي تقوم بها .  
ترشد الكلمة الدليلية Is الصنف ليقارن متغير الأختبار بالتعبير الذي يليها . تعرف الكلمة الدليلية To نطاقا من القيم .  
تستعمل البنية التالية الكلمتين الدليليتين Is و To والعديد من عوامل المقارنة لأختبار المتغير Age ولعرض احدى خمس رسائل:

Select Case Age

Case Is < 13

Label1.Caption = "Enjoy your youth!"

Case Is 13 To 19

Label1.Caption = "Enjoy your teens!"

Case 21

Label1.Caption = "You can drink wine with your meals."

Case Is > 100

Label1.Caption = "Looking good!"

Case Else

Label1.Caption = "That's a nice age to be .!"

End Select

شرح :

أذا كانت قيمة المتغير Age أصغر من 13 ، تظهر الرسالة "Enjoy your youth!" كما تظهر الرسالة "Enjoy your teens!" للأعمار التي تتراوح بين 13 و 19 ، الخ .  
أن بنية القرار Select Case هي أكثر وضوحا من البنية If...Then وأكثر فعالية عند اتخاذ ثلاثة قرارات متفرعة أو أكثر بناء على متغير واحد أو على خاصية واحدة .  
لكن عند إجراء مقارنتين أو أقل ، أو عند العمل على عدة قيم مختلفة ، قد تفضل استعمال البنية . If...Then

الى الدرس التالي:

### أستعمال الحلقات

ستتعلم في هذا الدرس كيفية تنفيذ كتلة جمل مرارا وتكرارا وذلك عن طريق أستعمال حلقة .  
ستستعمل حلقة For...Next لتنفيذ جملة عددا معيناً من المرات ، ثم ستستعمل حلقة Do لتنفيذ جمل الى أن يقيم التعبير الشرطي في الحلقة الى صح .  
كما ستتعلم كيفية أستعمال الطريقة Print لأظهار نصوص وأرقام على النموذج

- أستعمال حلقة For...Next
- أستعمال الطريقة Print
- أستعمال حلقة Do

### كتابة حلقات For...Next

تتيح لك الحلقة For...Next تنفيذ مجموعة معينة من الجمل البرمجية عددا محدداً من المرات في الاجراء الحداثي . هذا الامر مفيد اذا كنت تؤدي عدة عمليات حسابية متصلة ببعضها أو كنت تعمل على عناصر موجودة على الشاشة أو كنت تعالج عدة أجزاء من ادخال مستخدم . أن الحلقة For...Next هي مجرد طريقة مختصرة لكتابة لائحة طويلة من الجمل البرمجية .  
التركيب النحوي للحلقة For...Next يبدو كذلك :

For Variable = Start To End

Statements to be repeated

Next Variable

وفق هذا التركيب النحوي فإن الكلمات For و To و Next هي كلمات دلالية مطلوبة و = هو عامل مطلوب .  
مثلا ، تصدر الحلقة For...Next التالية أربعة أصوات تنبيه متتالية سريعة من مخرج الصوت في الكمبيوتر :

For I = 1 To 4

Beep

## Next i

هذه الحلقة المرادف الوظيفي لكتابة Beep أربع مرات في الاجراء . وهي بالنسبة الى المصرف مثل:

Beep

Beep

Beep

Beep

المتغير المستعمل في الحلقة هو I وهو حرف واحد يمثل ، حسب العادة ، العداد الرقمي الصحيح الاول في الحلقة For...Next . كلما تم تنفيذ الحلقة ، كلما ازداد المتغير العداد بواحد ( في المرة الاولى يحتوي المتغير على القيمة 1 وهي قيمة Start ، وفي المرة الاخيرة يحتوي على القيمة 4 ، أي قيمة End ) .

كما ستلاحظ في الامثلة التالية يمكنك الاستفادة الى حد بعيد من هذا المتغير في الحلقات .

## عرض المعلومات بأستعمال حلقة For...Next

أفتح مشروع جديد زد طول النموذج لأنشاء بعض المساحة الاضافية لأظهار الاخراج .

ضع زر أمر من الجهة اليمنى (Commandbox) .

أفتح إطار الخصائص ثم أضبط الخاصية Caption لزر الامر عند Loop .

أنقر النموذج مرة واحد للتبديل الى النموذج ، الآن تظهر خصائص النموذج في إطار الخصائص .

غير الخاصية Font الى Times New Roman .

تتحكم الخاصية Font بكيفية ظهور النص على النموذج . يمكنك أستعمال أي خط متوفر في

الكمبيوتر لكن خطوط التروتايب (TrueType) هي الافضل لأنه يمكن عرضها بعدة أحجام

ولأن شكلها على الشاشة هو نفسه عند الطباعة .

غير الخاصية AutoRedraw الى True .

أدا تم أخفاء نموذجك ، ستعيد الخاصية AutoRedraw أظهار نص تم عرضه بواسطة الطريقة

. Print

أنقر الزر Loop على النموذج نقرا مزدوجا .

يظهر الاجراء الحدتي Command1\_Click في إطار الشيفرة .

أكتب الجمل التالية في الاجراء

```
For I = 1 To 10
```

```
Print "Line"; i
```

```
Next i
```

تستعمل الحلقة For...Next هنا الطريقة لعرض الكلمة Line يليها عداد الحلقة 10 مرات

على النموذج .

توجه الفاصلة المقوطة (;) في الجملة Print فيجوال ببسك الى عرض المتغير العداد بجانب

السلسلة "Line" من دون ترك مسافة إضافية بينهما (لكنك رغم ذلك ستشاهد مسافة بين

"Line" وقيمة العداد عندما يشتغل البرنامج .

تلميح :

تدعم الطريقة Print أستعمال الفاصلة المنقوطة (;) والفاصلة (,) للفصل بين العناصر في لائحة

تعبيرها .

تباعد الفاصلة بين العناصر مسافة جدولة واحد ، جرب الطريقتين لتلاحظ الفرق .

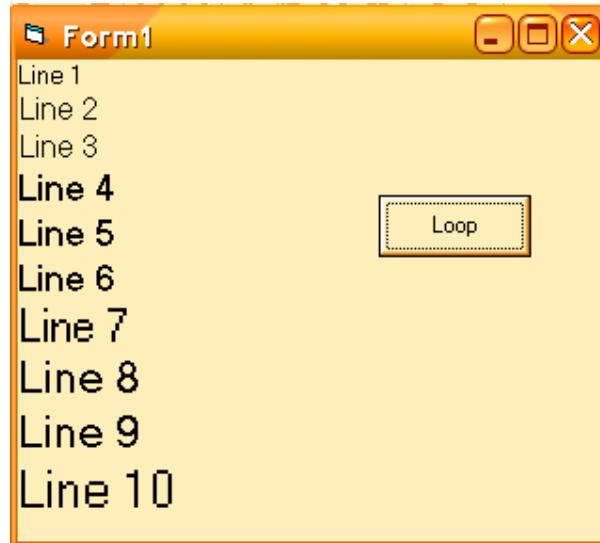
## تغيير خاصية في الحلقة For...Next



يتيح لك فيجوال بيسك تغيير الخصائص وتحديث المتغيرات المهمة في الحلقة .  
سنعدل البرنامج السابق بحيث يغير الخاصية FontSize في الحلقة For...Next .  
تحدد هذه الخاصية حجم الاحرف على النموذج .  
أفتح الاجراء الحدتي Command1\_Click في البرنامج السابق .  
أكتب الجملة التالية تحت الجملة For مباشرة:

```
For I = 1 To 10  
    FontSize = 10 + I  
    Print "Line"; I  
Next I
```

تضبط هذه الجملة الخاصية FontSize للنموذج عند 10 نقاط أكثر من قيمة عداد الحلقة .  
في المرة الاولى خلال الحلقة سيضبط حجم الخط عند 11 نقطة وفي المرة التالية عند 12 وهكذا  
حتى الحلقة الاخيرة عندما يتم تكبير الى 20 نقطة .  
بعد الانتهاء يجب أن تبدوا كالتالي :



كل مرة تم فيها زيادة عداد الحلقة ، أدى الى تكبير حجم الاحرف على النموذج .

### أنشاء حلقات For...Next معقدة

يمكن أن يكون المتغير العداد في الحلقة For...Next أداة مهمة في البرنامج . مع قليل من الخيال ، يمكنك أستعماله لأنشاء عدة تسلسلات من الارقام في الحلقات . لأنشاء حلقة فيها نمط عداد يختلف عن 1 ، 2 ، 3 ، 4 الخ ، يمكنك تحديد قيمة مختلفة لنقطة البداية Start في الحلقة ثم أستعمال الكلمة الدليلية Step لزيادة العداد عند كميات مختلفة مثلا ، الحلقة :

```
For I = 5 To 25 Step 5  
    Print I  
Next I
```

ستطبع تسلسل الارقام التالي على النموذج

```
5  
10  
15
```

20  
25

يمكنك تحديد قيم عشرية في الحلقة .مثلا :

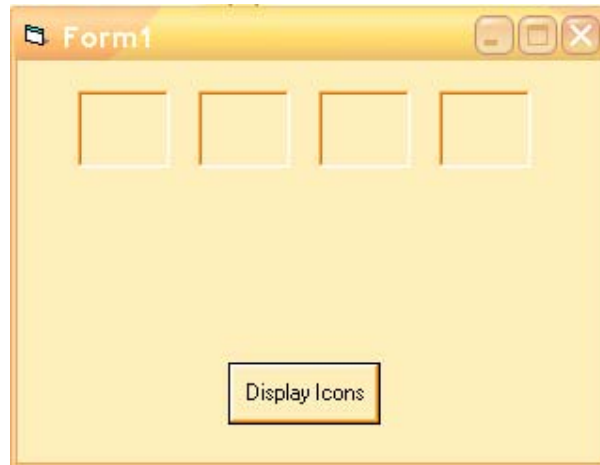
```
For I = 1 To 2.5 Step 0.5  
  Print I  
Next I
```

ستطبع الارقام التالية على النمودج :

1  
1.5  
2  
2.5

بالاضافة الى عرض المتغير العداد ، يمكنك استعمال العداد لضبط الخصائص أو لأحتساب قيم أو لمعالجة الملفات . يبين التمرين التالي كيف يمكنك استعمال العداد لفتح رموز فيجوال بيسك المخزنة على قوئك الثابت في ملفات تحتوي أسماؤها على أرقام . يبين البرنامج أيضا كيف يمكنك استعمال الحلقة For...Next للعمل على عدة كائنات صور بأعتبارها مجموعة واحدة . لتنظيم كائنات الصور بحيث يمكن معالجتها بفعالية .

- أفتح مشروع جديد .
- أنقر أداة التحكم صورة من مربع الادوات ثم أنشئ مربع صورة صغير بالقرب من الزاوية اليسرى .
- أختصر الامر < Edit Copy .
- أختصر الامر < Edit Paste .
- يعرض فيجوال بيسك رسالة تسألك أدا كنت ترغب بأنشاء مصفوفة تحكم (ControlArray) في البرنامج . مصفوفة التحكم هي مجموعة كائنات متماثلة في واجهة البرنامج . يحمل كل كائن في المجموعة نفس الاسم لذا يمكن أنتقاء المجموعة بكاملها وتعريفها دفعة واحدة . لكن يمكن استعمال كل كائن في المصفوفة على حدة مما يعطيك تحكما كاملا بكل بند من البنود الموجودة في واجهة البرنامج .
- أنقر Yes لأنشاء مصفوفة تحكم .
- أختصر الامر < Edit Paste مرة أخرى ثم أسحب الصورة الى يمين مربع الصورة الاولى الى أن كرر ما فعلنا الى أن يصبح عندنا أربع صور كالتالي :



بعد الانتهاء من اللصق ضع زر أمر CommandBox .  
الآن أضبط خصائصهم كالآتي :

الكائن	الخاصية	الاعداد
مصفوفة تحكم Image1	BorderStyle	1 – Fixed Single
مصفوفة تحكم Image1	Stretch	True
Command1	Caption	"Display icons"

- أنقر نفرا مزدوجا الزر Display icons على النموذج لعرض الاجراء الحداثي الخاص بكائن زر الامر .
- يظهر الاجراء الحداثي Command1\_Click في إطار الشيفرة .
- كبر إطار الشيفرة ثم أكتب الحلقة For...Next التالية :

For I = 1 To 4

Image1(I – 1).Picture = LoadPicture("c:\Vb6sbs\less07\misc0" & I & ".ico")

Next I

دعونا نشرح الكود :

الحلقة تستعمل الدالة LoadPicture لتحميل اربعة ملفات رموز من المجلد \Vb6sbs\less07\misc0 في قرصي الثابت . القسم الخدعة من الحلقة هو الجملة :

Image1(I – 1).Picture = LoadPicture("c:\Vb6sbs\less07\misc0" & I & ".ico")

التي تحمل الملفات من القرص الثابت . القسم الاول من الجملة :

Image1(I – 1).Picture

يستعمل الخاصية Picture لكل مربع صووة مصفوفة التحكم . يمكنك أستعمال البنود في مصفوفة التحكم من خلال فهارسها ، تستطيع ذكر مربعات الصور بشكل فردي في هذا المثال كالتالي : Image1(0) و Image1(1) و Image1(2) و Image1(3) .  
القم الموجود بين القوسين هو قيمة الفهرس في المصفوفة .  
في هذا المثال يتم أحتساب قيمة الفهرس الصحيح من خلال طرح 1 من المتغير العداد .  
يتم أنشاء أسم الملف بأستعمال المتغير العداد وعامل الضم (&).

LoadPicture("c:\Vb6sbs\less07\misc0" & I & ".ico")

تدمج أسم مسار وأسم والملحق .ico . لأنشاء أربعة أسماء ملفات رموز صالحة في القرص الثابت .

في هذا المثال يتم تحميل Misc01.ico و Misc02.ico و Misc03.ico و Misc04.ico في مربعات الصور من قرصي الثابت .

تنجح هذه الجملة لأن عدة ملفات المجلد \Vb6sbs\less07 مسماة على الشكل Miscxx.ico .

أن معرفة طريقة تسمية الملفات تتيح لك أنشاء حلقة For...Next مفيدة .  
مثلا آخر لتتوضح الفكرة .

ضع أداة Text Box ثم أنسخ الاداة وأصقها على الفورم ستخرج لك رسالة أنقر Yes أنسخ وأصق حوالي 5 أدوات Text Box .

ستلاحظ أن الادوات لها نفس الاسم فقط الاختلاف هو الفهرس كل أداة لها رقم فهرس فريد به

ضع زر أمر وضع الشيفرة التالية :

```

For I = 0 To 5
  If Text(I).Text <> "" Then
    Text(I).Text = "Iemad"
  End If
Next I

```

سأتركك أنت وعقلك لهذا الكود بالتوفيق أيدي تعبت والله .

### الجملة For...Next المركبة

فكرة البرنامج

كتابة برنامج يوضح عمل For...Next المركبة .

برنامج يقوم بطباعة رقم الحلقة الخارجية ورقم الحلقة الداخلية

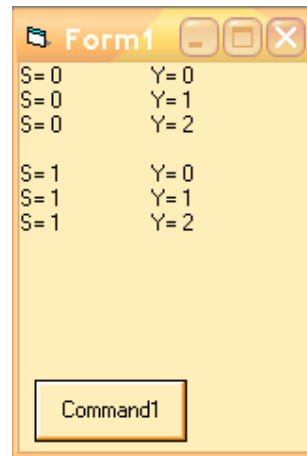
- أفتح مشروع جديد وضيف عليه زر أمر Command1
- أنقر زر الامر نقرأ مزدوجا يظهر لك محرر الشيفرة وأكتب التالي

```

For S = 0 To 1
  For Y = 0 To 2
    Print "S="; S, "Y="; Y
  Next
Print ""
Next

```

أخراج الشيفرة يجب أن يبدو كالتالي :



تكرر تنفيذ الاوامر الموجودة في الحلقة الخارجية بعدد مرات تكرر الحلقة الخارجية مرتين  
مثل طباعة سطر فارغ Print ""

يتكرر تنفيذ الاوامر الموجودة في الحلقة الداخلية بعدد مرات تكرر الحلقة الداخلية مضروبة في  
عدد مرات تكرار الحلقة الخارجية 6 مرات  
مثل الامر print "s=" ; s , "y=" ; y

### الجملة Exit For

تتيح لك الجملة Exit For الخروج من الحلقة For...Next قبل أن تنتهي تنفيذ كل تكراراتها .  
هكذا يصبح بإمكانك الاستجابة الى حدث معين حصل قبل أن تتم الحلقة العداد المحدد من  
المرات . مثلا:

```

For I = 1 To 10
  InpName = InputBox("Enter your name or Done")
  If InpName = "Done" Then Exit For
  Print InpName
Next I

```

هذه الجملة تحت المستخدم على كتابة عشرة أسماء وتطبعها على النموذج ، ألا أدا كتب الكلمة Done .  
 في هذه الحالة ينتقل البرنامج الى الجملة الاولى التي تلي الجملة Next .  
 تستعمل عادة Exit For مع الجملة If .

### كتابة حلقات Do

كبدل عن الحلقة For...Next يمكنك كتابة حلقة Do تنفذ مجموعة من الجمل الى أن تستوفي أحد الشروط في الحلقة . الحلقات Do مهمة لأنك غالبا لا تستطيع أن تعرف مسبقا كم مرة يجب أن تتكرر الحلقة .  
 مثلا ، قد ترغب في ترك المستخدم يكتب أسماء في قاعدة بيانات الى أن يكتب الكلمة Done في مربع الإدخال في تلك الحالة يمكنك استعمال حلقة Do لتتكرر من دون توقف حتى تتم كتابة السلسلة النصية "Done" .  
 للحلقة Do عدة تنسيقات ، وفقا لأين وكيف يتم تقييم شرطها . التركيب النحوي الاكثر شيوعا هو :

### Do While condition

Block of statements to be executed

### Loop

مثلا ، ستعالج الحلقة Do التالية الإدخال الى أن تتم كتابة الكلمة Done :

### Do While InpName <> "Done"

```

InpName = InputBox("Enter your name or Done to quit.")
If InpName <> "Done" Then Print InpName

```

### Loop

تكرر الحلقة طالما أن المتغير InpName لا يحتوي على الكلمة Done .  
 بمجرد كتابة الكلمة Done تتوقف الحلقة وينتهي الاجراء .  
 إذا أردت أن تستغل الحلقة دائما مرة واحد على الاقل في البرنامج ، ضع الاختبار الشرطي في أسفلها . مثلا :

### Do

```

InpName = InputBox("Enter your name or Done to quit.")
If InpName <> "Done" Then Print InpName

```

### Loop While InpName <> "Done"

في صلبها للحلقة Do السالقة ، لكنه يتم هنا تقييم شرط الحلقة بعد تلقي أسم من الدالة InputBox .  
 هذا مفيد في أنه يتم تحديث المتغير InpName قبل تقييم الاختبار الشرطي في الحلقة ، لذا تؤدي قيمة "Done" الموجودة سابقا الى تخطي الحلقة .

### تجنب الحلقة اللامتناهية

إذا لم يتم تقييم اختبار الحلقة الى خطأ أبدا ، سيتم تنفيذ الحلقة الى ما لا نهاية ولن يستجيب البرنامج الى الإدخال بعد ذلك . إليك المثال التالي :

Do

```
Number = InputBox("Enter a number to square . type -1 to quit.")  
Number = Number * Number  
Print Number
```

Loop While Number >= 0

في هذه الحلقة ، يكتب المستخدم رقما تلو الآخر ويقوم البرنامج بتربيعه وطبعه على النموذج . لسوء الحظ ، عندما يرغب المستخدم بإنهاء البرنامج لن يتمكن من ذلك لأن شرط الخروج المشار إليه لا يعمل .  
أد عندما يكتب المستخدم 1- يربعه البرنامج وتصبح قيمة المتغير Number تساوي 1 ( يمكن إصلاح هذه المشكلة من خلال وضع شرط خروج آخر )

### تحويل درجات الحرارة بأستعمال حلقة Do

- أفتح مشروع جديد
- أفتح إطار الخصائص ثم أضبط خاصية النموذج Visible عند False .
- عند ضبط الخاصية Visible لأحد النماذج عند False ، يخفي فيجوال بيسك النموذج عندما تشغل البرنامج بما أن هذا البرنامج يتلقى فقط درجات الحرارة بالفهرنهايت ويعيدها كدرجات مئوية فأن أخفاء نمودجه هو أمر جيد .
- أنقر النموذج نقرًا مزدوجًا .
- أكتب الجمل التالية بالحدث Form\_Load .

Prompt = "Enter a Fahrenheit temperature."

Do

```
FTemp = InputBox (Prompt, "Fahrenheit to celsius")  
If FTemp <> "" Then  
    Celsius = Int((FTemp + 40) * 5 / 9 - 40)  
    MsgBox (Celsius), , "Temperature in Celsius"  
End If
```

Loop While FTemp <> ""

End

تهتم هذه الاسطر التسعة بسألة الحسابات . يعين السطر الاول سلسلة نصية الى المتغير Prompt ، الذي سيستعمل بعد ذلك لعرض رسالة تعليمات في مربع الادخال . تطلب الحلقة Do بشكل متكرر من المستخدم أن يكتب درجة حرارة بالفهرنهايت وتحولها الى درجة مئوية ثم تعرضها على الشاشة بأستعمال الدالة MsgBox يتم تنفيذ الحلقة الى أن ينقر المستخدم الزر Cancel الذي يعيد قيمة فارغة ، أو خالية Null ، الى المتغير FTemp .  
أخيرا تفحص الحلقة القيمة الفارغة بأستعمال الاختبار الشرطي While في أسفل الحلقة . أخيرا تتولى الجملة البرمجية

Celsius = Int((FTemp + 40) \* 5 / 9 - 40)

عملية التحويل من فهرنهايت الى مئوية . تستعمل هذه الجملة صيغة تحويل قياسية ، لكنها تستعمل الدالة Int لتعيد الى المتغير Celsius قيمة لا تحتوي على منازل عشرية (يهمل كل ما هو موجود على يمين النقطة العشرية) . يضحى هذا الامر بعامل الدقة لكنه يساعدك على تجنب الحصول على أعداد طويلة مثل 21.1111111111 ، وهي القيمة المئوية للدرجة 70 فهرنهايت

شغل البرنامج أكتب 72 يتم تحويل درجة الحرارة من 72 الى 22 مئوية .

## أستعمال الكلمة الدليلية Until في حلقات Do

أن الحلقات Do التي عملت عليها حتى الآن أستعملت الكلمة الدليلية While لتنفيذ مجموعة من الجمل طالما أن شرط الحلقة بقي صح . يتيح لك فيجوال بيسك أيضا أستعمال الكلمة الدليلية Until في الحلقات Do ليتكرر تنفيذها الى أن يصبح الشرط صح . يمكن أستعمال الكلمة الدليلية Until في أعلى الحلقة Do أو في أسفلها لأختبار الشرط ، تماما كما هو الحال مع الكلمة الدليلية While . مثلا تستعمل الحلقة Do التالية الكلمة الدليلية Until ليتكرر الى أن يكتب المستخدم الكلمة Done في مربع ادخال :

Do

```
InpName = InputBox("Enter your name of Done.")
```

```
If InpName <> "Done" Then Print InpName
```

Loop Until InpName = "Done"

كما ترى ، الحلقة التي تستعمل الكلمة الدليلية Until تشبه كثيرا الحلقة التي تستعمل الكلمة الدليلية While ، بأستثناء أن شرط الاختبار يحتوي عادة على العامل المعاكس ، = (يساوي) بلا من <> (لا يساوي) في هذه الحالة .  
الفرق لتتوضح الامور أكثر الكلمة Until تعني ( لغاية ) أما الكلمة While تعني ( بينما )  
وشوف أنت وزوقك .  
وشكرا أدعولي .

DO جمل	ينفذ الجملة إذا كان الشرط صحيح يعيد تنفيذ الجملة
LOOP WHILE ( X < 4 ) DO WHILE ( X < 4 ) جمل	إذا كان الشرط صحيح ينفذ الجملة
LOOP Do جمل	ينفذ الجملة إذا كان الشرط خطأ يعيد تنفيذ الجملة
LOOP UNTIL ( X < 4 ) DO UNTIL ( X < 4 ) جمل	إذا كان الشرط خطأ ينفذ الجملة
LOOP	

الى الدرس التالي:

## المتغيرات والعوامل في فيجوال بيسك

ستتعلم في هذا الدرس كيفية أستعمال المتغيرات لتخزين البيانات مؤقتا في برنامجك وكيفية أستعمال العوامل الرياضية لتنفيذ مهام كالجمع والضرب .

كما ستتعلم كيفية استعمال الدالات الرياضية للقيام بحسابات تتضمن أرقاما ، وستستخدم الدالتين MsgBox و InputBox للحصول على معلومات وأظهارها بواسطة مربعات حوار .

- استعمال المتغيرات لتخزين البيانات في برامجك .
- الحصول على أدخل بأستعمال الدالة InputBox .
- عرض رسائل بأستعمال الدالة MsgBox .
- استعمال العوامل والدالات الرياضية في الصيغ .

### هيكلية جمل فيجوال بيسك البرمجية :

الجملة البرمجية هي أي تركيبة من كلمات فيجوال بيسك الدليلية ومن الخصائص والدالات والعوامل والرموز التي تنشئ معا تعليمة صالحة يتعرف عليها مصرف فيجوال بيسك . يمكن أن تكون الجملة البرمجية بأكملها مجرد كلمة دلالية بسيطة ، مثلا:

### Beep

التي تصدر نوتة من مكبر الصوت في كمبيوترك ، أو قد تكون تركيبة من عناصر ، كالجملة التالية تعين الوقت الحالي الى الخاصية Caption التابعة لعنصر التسمية :

Label1.	Caption	=	Time
أسم الكائن	أسم الخاصية	عامل التعيين	دالة فيجوال بيسك

تسمى قواعد البناء التي يجب أستعمالها عند بناء جملة برمجية (التركيب النحوي). يشارك فيجوال بيسك العديد من قواعد تركيبه النحوي مع إصدارات سابقة للغة البرمجة بيسك (Basic) ومع مصرفات لغات أخرى . الخدعة لكتابة جمل برمجية جيدة هي تعلم التركيب النحوي لأكثر عناصر اللغة فائدة ثم أستعمال تلك العناصر بشكل صحيح لمعالجة البيانات في برنامجك . لحسن الحظ أن فيجوال بيسك يتولى معظم الاعمال الصعبة عنك ، لذا سيكون الوقت الذي تصرفه في كتابة الشيفرة البرمجية قصيرا نسبيا ، وسيكون بالامكان أستعمال النتائج مجددا في برامج مستقبلية .

### أستعمال المتغيرات لتخزين المعلومات :

المتغير هو مكان تخزين مؤقت للبيانات في برنامجك . يمكنك أستعمال متغير واحد أو أكثر في الشيفرة ومن الممكن أن يحتوي المتغير على كلمات أو أرقام أو تواريخ أو خصائص . المتغيرات مفيدة لأنها تتيح لك إعطاء أسم قصير من السهل تذكره الى بيانات تعترزم العمل عليها . يمكن أن تحتوي المتغيرات على معلومات أدخلها المستخدم خلال تشغيل البرنامج أو على نتيجة عملية حسابية ما أو على قطعة بيانات تريد عرضها في نموذجك . بأختصار ، المتغيرات هي أدوات بسيطة يمكنك أستعمالها لتعقب أي نوع من المعلومات تقريبا .

### حجز مساحة للمتغيرات الجملة Dim

لتصريح المتغير علنيا قبل أن تستعمله ( عادة في بداية الاجراء الحداثي)، أكتب أسمه بعد الجملة Dim . مثال  
تنشئ الجملة التالية مساحة لمتغير يدعى LastName في أحد البرامج :

### Dim LastName

يمكنك تحديد نوع هذا المتغير أدا شئت ( ستتعلم المزيد عن أنواع البيانات الرئيسية لاحقا أن شاء الله ).

بعد تصريح المتغير كما سبق يمكنك تعيين له في الشيفرة . مثلا : تعيين الجملة البرمجية التالية الكنية Iemad الى المتغير LastName .



```
LastName = "iemad"
```

بعد هذا التعيين يمكن أستعمال المتغير LastName بدلا من الكنية iemad في الشيفرة . مثال :

```
Label1.Caption = LastName
```

عند أستعمال التالي في برنامجك ستظهر الكنية (iemad) بدلا من المتغير (LastName) لان المتغير ما هو الا تخزين أو إشارة الى بيانات معينة .  
جرب ذلك ضع أداة تسمية (label) ثم ضع التالي :

```
Private Sub Form_Load()
```

```
Dim LastName
```

```
LastName = "iemad"
```

```
Label1.Caption = LastName
```

```
End Sub
```

### تصريح المتغير من دون أستعمال Dim

يمكنك تصريح المتغير من دون أستعمال الجملة Dim وتسمى هذه العملية تصريحا ضمنيا .  
لتصريح متغير بهذه الطريقة ، أستعمل المتغير لوحده متجاهلا Dim كليا :

```
LastName = "iemad"
```

ملاحظه :

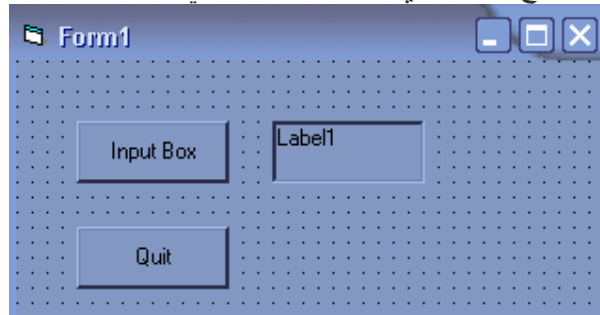
أدا قررت تصريح المتغيرات بأستعمال Dim دائما فقد ترغب بوضع الجملة Option Explicit في قسم التصاريح في نموذج بدء التشغيل لكل مشروع جديد .  
تستطيع القيام بهذا تلقائيا لكل مشاريعك الجديدة بأختيار الامر Options < Tools ثم انقر علامة التبويب Editor وانتقاء مربع الاختيار Require Variable Declaration .  
عندما تستعمل الجملة Option Explicit ، يعرض فيجوال بيسك رسالة خطأ كلما وجد متغير لم يتم تصريحه علنيا في الشيفرة (قد يكون السبب المحتمل هو خطأ في كتابة اسم المتغير).  
كنت قلقا بشأن الاخطاء الاملائية، ستساعدك هذه الجملة على تعقب أثرها واكتشافها .

### أستعمال متغيرات لتخزين الادخال

أحد الاستعمالات العملية للمتغير هو حفظ المعلومات التي يدخلها المستخدم . رغم أنه يمكنك في أغلب الاحيان أستعمال كائن ما ( كمربع سرد الملفات أو مربع نص ) لاستخراج تلك المعلومات، فقد ترغب أحيانا بالتعامل مع المستخدم مباشرة وحفظ الادخال في متغير وليس في خاصية .  
أحدى الطرق لفعل هذا هي بأستعمال الدالة Input Box لإظهار مربع حوار على الشاشة ثم تخزين النص الذي يكتبه المستخدم في أحد المتغيرات  
ستجرب الأسلوب في المثال التالي :

١ . أفتح مشروع جديد

٢ . ضع عليه زري أمر وتسمية كالتالي :



أنقر الزر الأول وضع الكود التالي :

Dim Prompt , FullName

Prompt = "Please enter your name."

FullName = InputBox(Prompt)

Label1.Caption = FullName

- أنت الآن تصرح متغيرين هما FullName و Prompt باستعمال الجملة Dim .
  - نعين في الجملة الثانية جملة نصية إلى المتغير Prompt .
  - سنستعمل هذه الرسالة كوسيلة نصية للدالة Input Box (الوسيلة هي قيمة أو تعبير يتم تمريره إلى إجراء فرعي أو دالة) .
  - يستدعي السطر الثالث الدالة Input Box ويعين نتيجة الاستدعاء (السلسلة النصية التي يكتبها المستخدم) إلى المتغير FullName .
  - Input Box هي دالة خاصة في فيجوال بيسك تعرض مربع حوار على الشاشة وتطلب من المستخدم كتابة شيء ما .
- شغل البرنامج وأنقر على الزر Input Box يخرج مربع حوار ، أكتب أسمك ثم Enter يظهر أسمك على كائن التسمية .  
القاعدة العامة هي التالي :

FullName	=	InputBox	(Prompt , Title)
أسم المتغير	عامل التعيين	اسم الدالة	الوسيطات

عرض رسالة باستعمال الدالة MsgBox .

- أنها تشرح نفسها من أسمها الواضح فلا داعي للشرح المطول ، أن هي ألا مربع رسالة تظهر للمستخدم جراء تفاعل ما أثناء تشغيل البرنامج ليوضح أمر ما للمستخدم أنت تحدد أن كان خطأ أو مساعدة أو شيء آخر دعنا نوضح كيف عملها .
- سنستعمل اختبارنا على البرنامج السابق ضيف زر أمر آخر إلى الفورم .
  - وأكتب الشيفرة التالية :

Dim Prompt

Prompt = "Please enter your name."

Msgbox (Prompt) , "Input Results"

- سنستدعي هذه الجملة الجديدة الدالة MsgBox ، وتعرض محتويات المتغير Prompt في مربع الحوار ، وتضع الكلمات Input Results في شريط العنوان .  
القاعدة العامة لها التالي :

MsgBox (Message, Number of Buttons, Title)

دعونا نشرح القاعدة العامة :

- الكلمة الأولى هي دالة في فيجوال بيسك .
  - الكلمة الثانية هي الرسالة التي ستخرج للمستخدم .
  - والثالثة هو شكل الرسالة التي تود أن تخرج بها للمستخدم وتحمل من 1 إلى العدد 5 .
  - وأما الأخيرة فهي عنوان الرسالة تظهر في شريط عنوان الرسالة .
- ضع هذا الكود في زر الأمر لترى المثال :

MsgBox ("Welcome Baby" , 1 , "رسالة ترحيب" , 1)

العمل مع أنواع معينة من البيانات

في معظم الحالات لن تحتاج إلا إلى نوع البيانات متنوع . يمكن للمتغيرات من هذا النوع أن يخزن كل أنواع بيانات فيجوال بيسك الرئيسية (المعرفة مسبقا) بالإضافة إلى بدالات التنسيق تلقائيا .  
من السهل استخدام المتنوعات ولن تجبرك على التفكير كثيرا بالحجم النهائي للمتغير عند تصريحه. لكن إذا أردت إنشاء شيفرة موجزة وسريعة قد تحتاج إلى استعمال أنواع بيانات محددة أكثر .  
مثلا اذا كان المتغير سيحتوي دائما على قيم رقمية صحيحة صغيرة (أرقام من دون منازل عشريه) ، يمكنك توفير مساحة الذاكرة عند تشغيل البرنامج من خلال تصريح المتغير كرقم صحيح وليس متنوع . وبما أن المتغير من النوع رقم صحيح سيسرع العمليات الحسابية فستكتسب بعض السرعة الإضافية عند استعماله.  
يسرد الجدول التالي أنواع البيانات الرئيسية في فيجوال بيسك . سترى في التمرين التالي كيف تعمل العديد من هذه الأنواع فلا تستعجل الأمور :

نوع البيانات	الحجم	النطاق	مثال عن أستعماله
Integer (عدد صحيح)	2 بايت	من -32,768 الى 23,767	Dim Iemad% Iemad = 37
Long (عدد صحيح طويل)	4 بايت	من -2,147,483,648 الى 2,147,483,647	Dim Iemad& Iemad = 350,000
Single (نقطة عائمة فردية الدقة)	4 بايت	من -3.402823E38 الى 3.402823E38	Dim Iemad! Iemad = 899.9
Double (نقطة عائمة مزدوجة النقطة)	8 بايت	من - الى 1.79769313486232D308 الى 1.79769313486232D308	Dim Iemad# Iemad = 3.1415926535
Currency (عملة)	8 بايت	من - الى 922337203685477.5808 الى 922337203685477.5807	dim Iemad @ Iemad = 7600300.50
String (سلسلة نصية)	بايت لكل حرف	من 0 الى 65,535 حرفا	Dim Iemad\$ Iemad = "Pointer"
Boolean (منطقي)	2 بايت	False أو True	Dim Iemad As Boolean Iemad = True
Date (تاريخ)	8 بايت	من 1 يناير 100 الى 31 ديسمبر 9999	Dim Iemad As Date Iemad = #3-1-63#
Variant (متنوع)	16 بايت للأرقام 22 بايت + 1 لكل حرف (للسلاسل)	كل نطاقات أنواع البيانات	Dim Iemad Iemad = 289.13

تلميح :-

كما نعرف فأننا لو أردنا تصريح متغير كرقم صحيح فأننا نصرحه كالاتي:

Dim I As Integer

تستطيع تصريح المتغير كالاتي بدلا من الجملة السابقة:

## Dim I%

هذا الرمز الذي آخر (%) المتغير يعني أن المتغير هو رقم صحيح أنها عادة قديمة في لغة البيسك لكن هناك بعض المبرمجين يستعملونها حتى الآن. بمعنى آخر بدل وضع (As Integer) هذا اختصار لها (%), أرجوا أن تكون واضحة.

### الثوابت (المتغيرات التي لا تتبدل)

إذا كان المتغير في برنامجك يحتوي على قيمة لا تتبدل أبداً (مثل ^، وهي قيمة رياضية ثابتة)، فقد ترغب في تخزين القيمة كثابت وليس كمتغير. الثابت هو أسم ذو معنى يحل محل رقم أو سلسلة نصية لا تتغير. الثوابت مفيدة لأنها تزيد من سهولة قراءة شيفرة البرنامج، كما أنها توفر من كمية الذاكرة المستعملة وتسهل تنفيذ التغييرات العامة. تعمل الثوابت كالمتغيرات إلى حد كبير لكنك لا تستطيع تغيير قيمها خلال التشغيل. يتم تصريحها بواسطة الكلمة الدلالية Const، كما هو مبين في المثال التالي:

### Const Pi = 3.14159265

تتشئ هذه الجملة ثابتاً يدعى Pi يحمل رقم عشري يمكن استعماله بدلاً من القيمة 3.141592665 مثال:

### If Number >= 20 Then Number = Number + Pi End If

إذا كان المتغير Number أكبر أو يساوي 20 أذن سنجمعه مع المتغير Pi الذي يحمل هذا الرقم 3.14159265 أنظر إلى السهولة لم أضطر إلى كتابة الرقم من جديد بل وضعت المتغير الذي صرحتة من قبل. وكلما أردت وضع هذا الرقم بدل كتابته من جديد أكتب المتغير Pi وهكذا.

### أستعمال عوامل فيجوال ببسك

الصيغة هي جملة تضم أرقاماً و متغيرات و عوامل و كلمات دلالية لإنشاء قيمة جديدة. يحتوي فيجوال ببسك على عدة عناصر مصممة للأستعمال في الصيغ. ستنمرن في هذا القسم على العمل مع العوامل الرياضية، أي على الرموز المستعملة لربط أقسام الصيغة فيما بينها. الرموز الرياضية التي ستستعملها هي التي تستعملها في الحياة اليومية وعملياتها بديهية إلى حد كبير، باستثناء بعض الحالات. سترى كل واحد من هذه الرموز موضحة في التمارين التالية يزود فيجوال ببسك العوامل التالية:

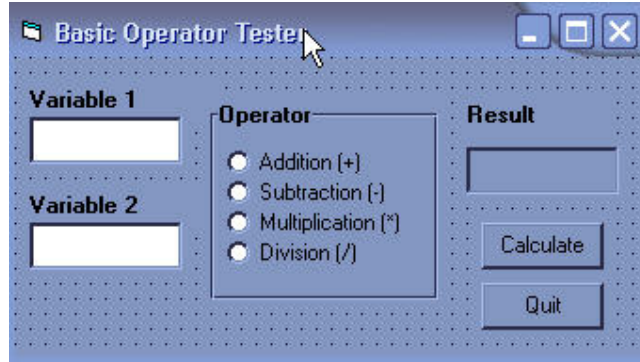
عامل التشغيل	العملية الرياضية
+	جمع
-	طرح
*	ضرب
/	قسمة
\	قسمة رقم صحيح
Mod	باقي القسمة
^	الأس (الرفع إلى قوة)
&	ضم السلاسل (تركيبية)

العوامل الرياضية الأساسية: + و - و \* و / و \  
أن عوامل الجمع والطرح والضرب والقسمة واضحة ويمكن استعمالها في أي صيغة تحتوي على أرقام أو على متغيرات رقمية. يوضح التمرين التالي كيف يمكنك استعمالها في برنامج فيجوال ببسك.

العمل مع العوامل الأساسية  
أفتح مشروع جديد وضع عليه التالي :  
عدد الاداة

Textbox = 2  
Option = 4  
Command = 2  
Label = 4  
Frame = 1

يجب أن يبدأوا مشروعك كهذا



مشروع آلة حاسبة مبسطة  
الاسماء واضحة لكل أداة لا داعي لاعادة تسميتهم أكتب أسمائهم كما موضح أعلاه

أنقر الفورم أي المشروع ليظهر لك محرر الكود وأكتب التالي :

```
()Private Sub Command1_Click  
Dim First, Second
```

```
First = Val(Text1.Text)  
(Second = Val(Text2.Text
```

```
If Option1.Value = True Then  
Label1.Caption = First + Second  
End If
```

```
If Option2.Value = True Then  
Label1.Caption = First - Second  
End If
```

```
If Option3.Value = True Then  
Label1.Caption = First * Second  
End If
```

```
If Option4.Value = True Then
```

```
Label1.Caption = First / Second
End If
End Sub
```

```
()Private Sub Command2_Click
End
End Sub
```

دعونا نشرح الكود بشئ من التفصيل :

تصرح الجملة الاولى متغيرين للاستعمال العام من النوع متنوع يعني أننا لم نكتب نوع المتغير لذلك فيجوال ببسك سوف يعتبره من النوع متنوع أي (Variant) سيستعمل هذان المتغيران لتخزين القيم المكتوبة في مربعي النص ، وستكون عندهما المرونة الكافية لمعالجة أي نوع من البيانات الرقمية تريد أستعماله .

تحمل الجملتان التاليتان البيانات من مربعات النص الى المتغيرات وتحول السلاسل النصية الى أرقام بأستعمال الدالة (Val) .

الدالة (Val) هي روتين خاص يحول وسيطة نصية الى قيمة رقمية . هذا التحويل ضروري لصحة سير عملية الجمع في هذا البرنامج .

أن نوع البيانات الافتراضي الذي يعيده النص هو نص .

هذه ليست بمشكلة بالنسبة لثلاثة عوامل .

تعمل العوامل - و \* و / مع الارقام فقط ، لذا عندما ينتقي المستخدم أحد هذه العوامل في البرنامج ، يحول فيجوال ببسك تلقائيا القيم المعادة الى المتغيرين (First , Second) الى أرقام .

يعمل العامل + مع السلاسل النصية والارقام على حد سواء . بما أن نوع البيانات الافتراضي الذي يعيده مربع النص هو نص ، سيعتبر فيجوال ببسك القيمتين (First , Second) أنهما نص عندما تستعمل العامل + .

عندها سيدمج ، أو يضم ، القيمتين بدلا من جمعهما مع بعضهما حسابيا مثلا:

$$100 + 17 = 117$$

نفهم من هذا كله أننا لو لم نستعمل (Val) سيتم ضم السلاسل بدل جمعهما ، جرب ذلك وأجمع عددين مع بعض من دون (Val) .

دعونا نشرح الكود التالي :

```
If Option1.Value = True Then
Label1.Caption = First + Second
End If
```

أذا كانت الخاصية (Value) لزر الخيار الاول تساوي (True) (اذا تم نقر الزر) يتم عندها جمع المتغيرين بواسطة العامل + وتعين النتيجة الى التسمية (Label1) تملك الصيغ الثلاث المتبقية المنطق نفسه أذ يستعمل كل منها بنية قرار If...Then والخاصية (Caption) للكائن (Label1) .

### دالات فيجوال ببسك الرياضية

قد ترغب من حين الى آخر أجراء بعض التعديلات في برامجك كتحويل قيمة الى نوع آخر أو احتساب نتيجة تعبير رياضي معقد أو اعتماد مبدا العشوائية في برامجك .

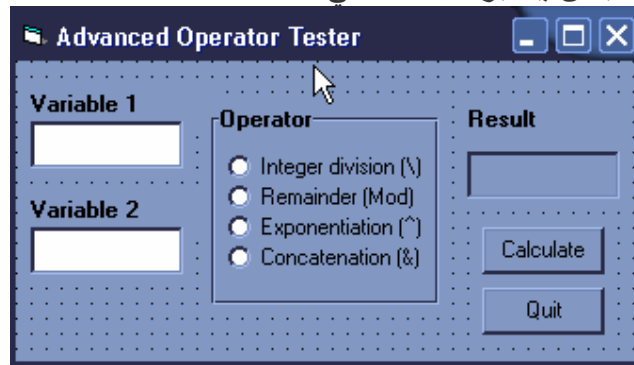
تتيح لك دالات فيجوال ببسك التالية العمل على الارقام في الصيغ .

وكما هو الحال مع أي دالة أخرى ، يجب أن تستعمل الدالات الرياضية في جملة برمجية ، وهي ستعيد قيمة الى البرنامج . في الدوال التالية تمثل الوسيطة n الرقم أو المتغير أو التعبير الذي تريد ان تقيمه الدالة .

الدالة	وضيفتها
Abs(n)	تعيد القيمة المطلقة ل n
Atn(n)	تعيد قوس مماس n ، بالراديان .
Cos(n)	تعيد جيب تمام n . تقاس الزاوية بالراديان .
Exp(n)	تعيد الثابت e مرفوعا الى الاس n .
Rnd(n)	تنتج رقما عشوائيا بين 0 و 1 .
Sgn(n)	تعيد -1 إذا كان n أقل من صفر و 0 إذا كان n صفرا و +1 إذا كان أكبر من صفر
Sin(n)	تعيد جيب زاوية n . تقاس الزاوية n براديان .
Sqr(n)	تعيد الجذر التربيعي ل n .
Str(n)	تحول قيمة رقمية الى سلسلة .
Tan(n)	تعيد مماس n . تقاس الزاوية n بالراديان .
Val(n)	تحول قيمة سلسلة الى رقم .

#### استعمال العوامل المتقدمة : \ و Mod و ^ و &

بالإضافة الى العوامل الرياضية الاساسية الاربعة ، يتضمن فيجوال بيسك أربعة عوامل متقدمة تؤدي قسمة العدد الصحيح (\) وباقي القسمة (Mod) والأس (^) وضم السلاسل (&).  
 هذه العوامل مفيدة في الصيغ الرياضية ذات الاستعمال الخاص وفي برامج معالجة النصوص .  
 يبين لك البرنامج التالي (هو تعديل طفيف على البرنامج السابق) كيف يمكنك أستعمال كل واحد من هذه العوامل .  
 يجب أن يبدأ برنامجك كالتالي :



لاحظ أننا لم نغيّر الكثير ، فقط غيرنا تسميات أزرار الاختيار .  
 أنقر على المشروع نقرتين مزدوجتين ليظهر محرر الكود وضع الشيفرة التالية داخل المحرر .

```
Private Sub Command1_Click()
```

```
Dim First, Second
```

```
First = Text1.Text
```

```
Second = Text2.Text
```

```
If Option1.Value = True Then
    Label1.Caption = First \ Second
End If
```

```
If Option2.Value = True Then
    Label1.Caption = First Mod Second
End If
```

```
If Option3.Value = True Then
    Label1.Caption = First ^ Second
End If
```

```
If Option4.Value = True Then
    Label1.Caption = First & Second
End If
End Sub
```

```
Private Sub Command2_Click()
    End
End Sub
```

- أكتب ٩ في مربع النص Variable 1
- ثم أكتب بالمربع النص التالي Variable 2
- أنقر زر الخيار Integer Division ثم الزر Calculate
- يتم الآن تطبيق العامل على القيمتين ويظهر الرقم 4 في المربع Result .  
تعطي القسمة الصحيحة فقط العدد الكامل الناتج عن عملية القسمة . مع أن قسمة 9 على 4 تعطي 4.5 ،  
ألا أن القسمة العددية الصحيحة تعطي فقط الجزء الصحيح الكامل (الرقم 4) .  
قد تجد هذا الأمر مفيداً إذا كنت تعمل على كميات ليس من السهل قسمتها إلى أجزاء كسرية ، كعدد البالغين الذين يتسعون في السيارة مثلاً .
- أنقر الخيار Remainder ثم الزر Calculate .  
يظهر الرقم 1 في المربع Result . يعيد عامل باقي القسمة (القسم المتبقي الذي لا يمكن قسمته ) بعد قسمة العددين . بما أن قسمة 9 على 2 تعطي 4 ويكون الباقي 1 (2\*4+1=9) فإن النتيجة التي يعطيها العامل Mod هي 1 .  
يتيح لك العامل Mod تتبع "ما تبقى" من العمليات الحسابية ، كقيمة الفكة المتبقية بعد حدوث عملية مالية .
- أنقر الخيار Exponentiation ثم الزر Calculate .  
يظهر الرقم 81 في مربع النص Result .  
يرفع عامل الالاس (^) العدد إلى أس نفسه .

بما أن  $9^2$  يساوي 81 ، فإن النتيجة التي يعطيها العامل  $^$  هي 81 .



في صيغة فيجوال ببسك ، يكتب  $9^2$  بهذا الشكل  $9^2$  .

- أنقر الزر Concatenation ثم الزر Calculate .  
تظهر السلسلة "92" في المربع Result . عامل ضم السلاسل (&) يدمج السلسلتين المحددتين في الصيغة .  
لا تكون النتيجة ("92" في هذه الحالة) عددا ، بل تركيبة من الحرف "9" والحرف "2" .  
يمكن تنفيذ ضم السلاسل فقط على المتغيرات النصية والسلاسل التي تحيط بها علامات اقتباس والمتغيرات التي من النوع متنوع .  
بما أن المتغير المستعمل في هذا البرنامج هي من النوع متنوع ، يتم تحويلها تلقائيا الى نصوص لأتمام العملية .  
لرؤية كيف يعمل هذا العامل مع الاحرف ، أكتب بعض الكلمات في مربع النص ثم أنقر الزر Calculate مرة ثانية .  
لا داعي لشرح الكود لانه يعمل مثل سابقة في البرنامج السابق .

### أولوية العوامل

لقد عملنا في التمرينين السابقين على سبعة عوامل رياضية وعامل سلاسل واحد .  
يتيح لك فيجوال ببسك وضع قدر ما تشاء من العوامل الرياضية في صيغة واحدة طالما كان هناك عامل واحد بين كل متغير رقمي وتعبير رقمي .  
مثلا هذه صيغة مقبولة في فيجوال ببسك :

$$\text{Total} = 10 + 15 * 2 / 4 ^ 2$$

تعالج هذه الصيغة عدة قيم وتعين النتيجة الى متغير اسمه Total . لكن كيف يتم تقييم تعبير كهذا في فيجوال ببسك ؟  
بمعنى آخر ، ما هي العوامل الرياضية التي يقيمها فيجوال ببسك أولا عند حل الصيغة ؟ ربما لم تلاحظ ذلك ، لكن ترتيب التقييم له تأثير كبير في هذا المثال .  
يحل فيجوال ببسك هذه المعضلة بأعماده ترتيبا معيناً لأولويات العوامل الرياضية .  
لائحة القواعد تلك تخبر فيجوال ببسك عن العوامل التي يجب أن يستعملها أولا عند تقييمه تعبيراً يحتوي على أكثر من عامل واحد  
يسرد الجدول التالي كل العوامل حسب الترتيب الذي سيتم تقييمها فيه (يتم تقييم العوامل الموجودة في المستوى نفسه في هذا الجدول من اليسار الى اليمين مثلما تبدوا في التعبير) .

العامل (العوامل)	ترتيب الاولوية
()	يتم تقييم القيم الموجودة بين قوسين أولا دائما .
^	يأتي الأس (رفع عدد الى قوة) ثانيا .
-	يأتي أنشا عدد سالب ثالثا .
/*	يأتي الضرب والقسمة رابعا .
\	تأتي القسمة العددية الصحيح خامسا .
Mod	تأتي باقي القسمة سادسا .
+ -	يأتي الجمع والطرح أخيرا .

أطلاقا من ترتيب الاولويات المبين هنا ، سيعالج فيجوال ببسك التعبير التالي  
 $\text{Total} = 10 + 15 * 2 / 4 ^ 2$   
بأستعمال الخطوات التالية ( لقد الخط المائل والازاهي لأظهار ترتيب التقييم والنتيجة ) .

$$\text{Total} = 10 + 15 * 2 / 4 ^ 2$$

$$\text{Total} = 10 + 15 * 2 / 16$$

$$\text{Total} = 10 + 30 / 16$$

$$\text{Total} = 10 + 1.875$$

$$\text{Total} = 11.875$$

خطوة متقدمة : أستعمال الاقواس في الصيغة .

يمكنك أستعمال زوج من الاقواس أو أكثر في الصيغة لتوضيح ترتيب الاولوية . مثلا ، سيحتسب فيجوال ببسك نتيجة الصيغة

$$\text{Number} = (8 - 5 * 3) ^ 2$$

من خلال أحتساب القيمة الموجودة بين قوسين (-7) قبل الاس - رغم أن الاس يحتل مرتبة أعلى ممن الطرح والضرب في ترتيب الاولويات . يمكنك تحسين الاحتماب أكثر عن طريق وضع أقواس متداخلة في الصيغة .  
مثلا ، الصيغة التالية :

$$\text{Number} = ((8 - 5) * 3) ^ 2$$

تجعل فيجوال ببسك يحتسب الفارق في القوسين الداخليين أولا ثم يحتسب القوسين الخارجييين ، وأخيرا الاس النتيجة التي تعطيها الصيغتان مختلفة : نتيجة الصيغة الاولى هي 49 بينما الثانية 81 .

تستطيع الاقواس تغيير نتيجة العملية الرياضية ، بالإضافة الى أنها تسهل قرائتها .  
أرجوا أن تكونوا فهمتم المتغيرات على أنواعها .

---

وأخيرا لمن أستفاد من هذه الدروس أن يدعي لي من قلبه وأي أستفسار أو ملاحظه الرجاء ارسالها على العنوان التالي :

[Iemad2@hotmail.com](mailto:Iemad2@hotmail.com)

أنا وضحت أول الكتاب أنني أريد كتابة دروس عن قواعد البيانات ( خاصة ) ممكن وانت تقرأ هذه الكلمات أكون قد نشرت هذا الكتاب ، وأذا أردت كتاب قواعد البيانات راسلني بينما أكون قد كتبته .