



وزارة التربية

تقنية المعلومات

للمصف الثاني عشر
الجزء الأول

Microsoft
Visual Basic 2008
Express Edition



visual Basic

المرحلة الثانوية
الطبعة الثانية

12



..... اسم الطالب :

..... الفصل :

..... رقم الجهاز :



وزارة التربية

تقنية المعلومات

للمصف الثاني عشر

الجزء الأول

تأليف

أ.مكي فاخر الخباز أ. محمد عبد الغني أحمد

أ. مريم مختار كابلي أ. عادل جمعة عبد الجيد

أ. محمد السيد إبراهيم أ. أحمد محمد عيسى

أ. محمود عبدالمفدي أبو النجا

Visual Basic

الطبعة الثانية

١٤٣٩ / ١٤٤٠ هـ

٢٠١٨ / ٢٠١٩ م

حقوق التأليف والطبع والنشر محفوظة لوزارة التربية - قطاع البحوث التربوية والمناهج
إدارة تطوير المناهج

12

الطبعة الأولى ٢٠٠٨ / ٢٠٠٩ م

الطبعة الثانية ٢٠١١ / ٢٠١٢ م

٢٠١٣ / ٢٠١٤ م

٢٠١٥ / ٢٠١٦ م

٢٠١٧ / ٢٠١٨ م

٢٠١٨ / ٢٠١٩ م

لجنة مواءمة المرحلتين المتوسطة والثانوية

أ. نجيبة أحمد دشتي

أ. محمد السيد إبراهيم

أ. محمد عبد الغني أحمد

أ. إيمان عبدالعزيز الفارسي

أ. محمد عبدالجواد الخليجي

أ. محمد جابر موسى

أ. حسام فتحي سليمان

أ. حنان علي غضنفري

أ. أحمد السيد الحسيني

أ. منى محمد أحمد الكندري

مطابع الألفين

أودع بمكتبة الوزارة تحت رقم (٥٨) بتاريخ ٥/٥/٢٠١١ م





صاحب السمو الشيخ أحمد الجابر الصباح
أمير دولة الكويت



سَيِّدُ الشَّيْخِ نَوَافِ بْنِ أَحْمَدَ بْنِ جَابِرِ بْنِ الصَّبَّاحِ
وَلِيِّ عَهْدِ دَوْلَةِ الْكُوَيْتِ

المحتوى

١١ المقدمة
الباب الأول : مدخل إلى البيسك المرئي	
١٧ الفصل الأول : مفاهيم أساسية
٢٥ الفصل الثاني : واجهة البيسك المرئي ٢٠٠٨
الباب الثاني : مراحل بناء برنامج	
٣٧ الفصل الأول : بداية بناء برنامج
٤٥ الفصل الثاني : ضبط عناصر البرنامج
٥٥ الفصل الثالث : كتابة التعليمات البرمجية
الباب الثالث : التعليمات البرمجية الأساسية	
٦٩ الفصل الأول : المتغيرات والثوابت
٨٧ الفصل الثاني : التفرع
٩٥ المشاريع
١٠١ كراس المعلم
١٢٣ التقويم
١٥٩ المراجع



المقدمة

عزيزي المتعلم :

أنت تخطو خطواتك الأخيرة لإنهاء الدراسة بالمرحلة الثانوية ، وتستعد لتخطو خطواتك الأولى لبدء دراستك الجامعية، وتحمل معك العديد من المعلومات التي اهتم المعنيون بتأليف مناهج الحاسوب على أن تكون مترابطة ومتكاملة ، بدءاً من الصف الأول إلى الصف الثاني عشر .

فقد درست خلال السنوات السابقة العديد من المهارات الأساسية للحاسوب، وتعرفت على مكوناته المادية والبرامج التطبيقية من معالجات نصوص وجداول حسابية وعروض تقديمية وبرامج تصميم مواقع الإنترنت وقواعد بيانات .

وحرصاً منا على أن تكمل دراستك بالمرحلة الثانوية بتعلم إحدى لغات البرمجة السهلة، ستدرس في هذا الصف لغة (بيسك المرئي Visual Basic)، لما لها من أهمية في دراستك بالمرحلة الجامعية ، وحتى بعد تخرجك إلى الحياة العملية.

ومن هذا المنطلق نضع بين يديك الجزء الأول من كتاب «تقنية المعلومات للصف الثاني عشر»، والذي يتكون من ثلاثة أجزاء:

الباب الأول: مفاهيم أساسية

فيه سنتعرف على مكونات الكيان البرمجي ، وأهمية لغات البرمجة ، وأهمية لغة بيسك المرئي، ومكونات واجهة البيسك المرئي وكيفية التعامل مع هذه المكونات .

الباب الثاني: بناء برنامج

وفيه سنتعرف على كيفية تحليل المشكلات، وخطوات إنشاء مشروع، وكتابة التعليمات البرمجية، وترجمة المشروع إلى برنامج تنفيذي.

الباب الثالث: تعليمات برمجية أساسية

وفيه سنتعرف على بعض العمليات الحسابية وتخصيص القيم والتعامل مع المتغيرات .
أملين أن يحقق هذا الكتاب الهدف منه، وأن يكون ما فيه من معلومات اللبنة الأساسية في تعلمك لغات البرمجة ، لتتفج بها وطنك ومن حولك .

لجنة المواءمة





الباب الأول

مدخل إلى البيسك المرئي

الفصل الأول

مفاهيم أساسية

الفصل الثاني

واجهة البيسك المرئي ٢٠٠٨م



visual Basic



الفصل الأول

مفاهيم أساسية

تعريف البرنامج VB

تعريف لغة البرمجة VB

أهمية لغات البرمجة VB

أقسام لغات البرمجة VB



visual Basic



كلنا ندرك أهمية البرامج بالنسبة للحاسوب، فبدون البرامج المختلفة لا نستطيع الاستفادة من إمكانيات الحاسوب المادية، تساعدنا لغات البرمجة على بناء البرامج بجميع أنواعها، وقبل دراسة لغات البرمجة لا بد أن نتعرف على المفاهيم التالية.

أولاً: تعريف البرنامج

هو مجموعة من التعليمات المنطقية التي يوجه المبرمج بها الحاسوب للقيام بمهمة محددة كإجراء عمليات حسابية أو طباعة تقارير أو رسم مخططات بيانية ... إلخ.



يسمى الشخص الذي يقوم بكتابة تعليمات برامج الحاسوب باسم «مبرمج

الحاسوب Programmer» .



ثانياً: تعريف لغة البرمجة

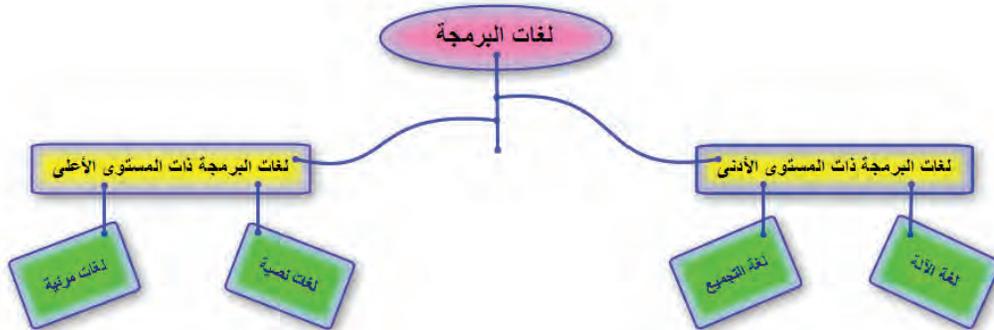
لغة البرمجة هي أداة تساعدنا على بناء البرامج بجميع أنواعها لمختلف الأغراض .

ثالثاً: أهمية لغات البرمجة

- يعتبر تعلم لغات البرمجة من أهم المواد التعليمية في الحاسوب والتي تكمن فوائدها في الآتي :
- 1- الاستقلالية في بناء البرامج الخاصة على وجه الخصوص هدف وطني لجميع دول العالم لضمان سرية المعلومات الوطنية المهمة.
 - 2- البرامج الخاصة توفر الاحتياجات المحلية المتنامية للمجتمع التي لا تلبها في الغالب البرامج العامة.
 - 3- إعداد الكفاءات البشرية لصناعة البرمجيات حيث أصبحت صناعة البرمجيات من أهم مصادر الدخل لبعض الدول.

رابعاً: أقسام لغات البرمجة

الشكل التالي يوضح أقسام لغات البرمجة :



(١) لغات برمجة ذات المستوى الأدنى :

سميت بذات المستوى الأدنى لبعدها عن لغة الإنسان أو لقربها من لغة الآلة. وتنقسم إلى نوعين :

أ- لغة الآلة Machine Language

هي اللغة الأساسية لمكونات الحاسوب المادية، وتعتمد على أرقام النظام الثنائي (١،٠) في كتابة التعليمات البرمجية التي يوجهها المبرمج للحاسوب.

تعد لغة الآلة من أصعب لغات البرمجة على الإنسان ، لأنه لا يستطيع حفظ هذا السيل من التركيبات الرقمية لرقمين (١٠٠) ، فتخيل لو أن لغتك التي تتحدث بها مع الآخرين تتكون من حرفين فقط حرف الألف والباء!!



أنا جدول أسهل
حفظ أرقام
لغة الآلة

0000	=	0
0001	=	1
0010	=	2
0011	=	3
0100	=	4
0101	=	5
0110	=	6
0111	=	7
1000	=	8
1001	=	9
1010	=	10
1011	=	11
1100	=	12
1101	=	13
1110	=	14
1111	=	15
10000	=	16
10001	=	17
10010	=	18
10011	=	19
10100	=	20
10101	=	21
10110	=	22
10111	=	23
11000	=	24
11001	=	25
11010	=	26
11011	=	27
11100	=	28
11101	=	29
11110	=	30
11111	=	31
100000	=	32

أرقام
لغة الآلة

أرقام
عشرية

ولكن حتى
لو حفظت سأحتاج
وقتاً كبيراً جداً لكتابة
جملة واحدة بهذه اللغة



مبرمج
الحاسوب

ب - لغة التجميع Assembly Language

هي لغة تعتمد على اختصارات مثل (Add ، Int ... إلخ) بالإضافة إلى الأرقام في كتابة التعليمات البرمجية التي يوجهها المبرمج للحاسوب ، وتحتاج لمترجم interpreter يترجمها إلى لغة الآلة .

أنا جدول أسهل
حفظ تعليمات
العمليات الحسابية
بلغة التجميع

التعليمات	الرمز	مثال
ADD	Add	Dest=Dest+Source
ADC	Add with Carry	Dest=Dest+Source+CF
SUB	Subtract	Dest=Dest-Source
SBB	Subtract with borrow	Dest=Dest-Source-CF
DIV	Divide (unsigned)	Op:byte AL=AX / Op AH=Rest
DIV	Divide (unsigned)	Op:word AX=DX:AX / Op DX=Rest
DIV	Divide (unsigned)	Op:double EAX=EDX:EAX / Op EDX=Rest
IDIV	Signed Integer Divide	Op:byte AL=AX / Op AH=Rest
IDIV	Signed Integer Divide	Op:word AX=DX:AX / Op DX=Rest
IDIV	Signed Integer Divide	Op:double EAX=EDX:EAX / Op EDX=Rest
MUL	Multiply (unsigned)	Op:byte AX=AL*Op # AH=0
MUL	Multiply (unsigned)	Op:word DX:AX=AX*Op # DX=0
MUL	Multiply (unsigned)	Op:double EDX:EAX=EAX*Op # EDX=0
IMUL	Signed Integer Multiply	Op:byte AX=AL*Op # AL sufficient
IMUL	Signed Integer Multiply	Op:word DX:AX=AX*Op # AX sufficient
IMUL	Signed Integer Multiply	Op:double EDX:EAX=EAX*Op # EAX sufficient
INC	Increment	Op=Op+1 (Carry not affected)
DEC	Decrement	Op=Op-1 (Carry not affected)
CMP	Compare	Op1-Op2
SAL	Shift arithmetic left (=SHL)	
SAR	Shift arithmetic right	
RCL	Rotate left through Carry	
RCR	Rotate right through Carry	
ROL	Rotate left	
ROR	Rotate right	

ما شاء الله
كل هذه التعليمات
المختصرة . تحتاج لذاكرة
فولاذية لتذكرها
!!!

ميربح
الحاسوب

(٢) لغات البرمجة ذات المستوى الأعلى :

سميت بذات المستوى الأعلى لقربها من لغة الإنسان .

وتنقسم إلى نوعين :

أ - لغات البرمجة النصية Script Languages

هي لغات تعتمد على عبارات إنجليزية اصطلاحية للدلالة على الأوامر التي توجه الحاسوب .

تعتبر لغات البرمجة النصية أسهل بكثير من لغة الآلة والتجميع لأن أوامرها تشبه لغة الإنسان



من أشهر اللغات النصية لغة البيسك «BASIC» البسيطة التي تناسب المبتدئين في مجال البرمجة، ولغة السي «C» التي تناسب المتخصصين في بناء برامج متقدمة مثل نظم التشغيل وبرامج التطبيقات.

الشكل التالي يوضح برنامجاً تم بناؤه بإحدى لغات البرمجة النصية، وهو مكتوب بإحدى إصدارات لغة بيسك السابقة التي كان يطلق عليها "Quick Basic"، والذي يطلب من المستخدم سعر سلعة ما ثم يحسب سعر البيع للسلعة .

شاشة البرنامج أثناء التصميم

```

E:\VISUAL~1\intro\disk1\Me...
File Edit U E:\VISUAL~1\intro\disk1\Me
10 INPUT n
20 n = n * .5
30 p = n + n
40 PRINT "price="; p
50 END
    
```

٢ يحسب هامش الربح

٤ يطبع على الشاشة سعر البيع

١ يطلب من المستخدم سعر الشراء

٣ يحسب سعر البيع

٥ نهاية البرنامج

شاشة البرنامج أثناء التنفيذ

```

E:\VISUAL~1\intro\disk1\Me...
? 50
price= 75
    
```

٢ يطبع البرنامج سعر البيع

١ يكتب المستخدم سعر الشراء

لغات البرمجة النصية سهلة في كتابة أوامرها للمبرمج، ولكنها غير مألوفة لمستخدمي البرامج البسطاء.



ب - لغات البرمجة المرئية

هي لغات تبني برامج ذات واجهة رسومية تفاعلية سهلة لمستخدمي الحاسوب، ومن خلال تضمينها لأوامر نصية توجه الحاسوب للمطلوب.

الشكل التالي يوضح نفس البرنامج النصي السابق ذكره في لغات البرمجة النصية، ولكن بلغة بيسك المرئي «Visual Basic».



- ١ - إدخال سعر الشراء داخل المتغير في الذاكرة.
- ٢ - الضغط على «احسب سعر البيع» من أجل المعالجة.
- ٣ - تنفيذ معادلة سعر البيع عبر الذاكرة.
- ٤ - طباعة الناتج في مربع سعر البيع.

الفصل الثاني

واجهه البيسك المرئي ٢٠٠٨ م

٢٠٠٨ تعريف البيسك المرئي VB

٢٠٠٨ مزايا البيسك المرئي VB

٢٠٠٨ تشغيل البيسك المرئي VB

٢٠٠٨ فتح مشروع سبق إنشاؤه VB

٢٠٠٨ عناصر واجهه البيسك المرئي VB

٢٠٠٨ التحكم في نوافذ الواجهه VB



visual Basic



تعتبر لغة البيسك المرئي Visual Basic 2008 من أسهل لغات البرمجة التي يمكن من خلالها إنشاء التطبيقات التي تعمل تحت نظام النوافذ وتطبيقات الإنترنت، وكذلك برمجة تطبيقات أجهزة الهاتف المحمول وأجهزة الحاسوب الكفية .



أولاً : تعريف البيسك المرئي ٢٠٠٨

هي أداة تطوير يمكن من خلالها بناء تطبيقات متنوعة تستجيب للأحداث لنظم تشغيل النوافذ أو مواقع الإنترنت أو أجهزة الحاسوب الكفية .

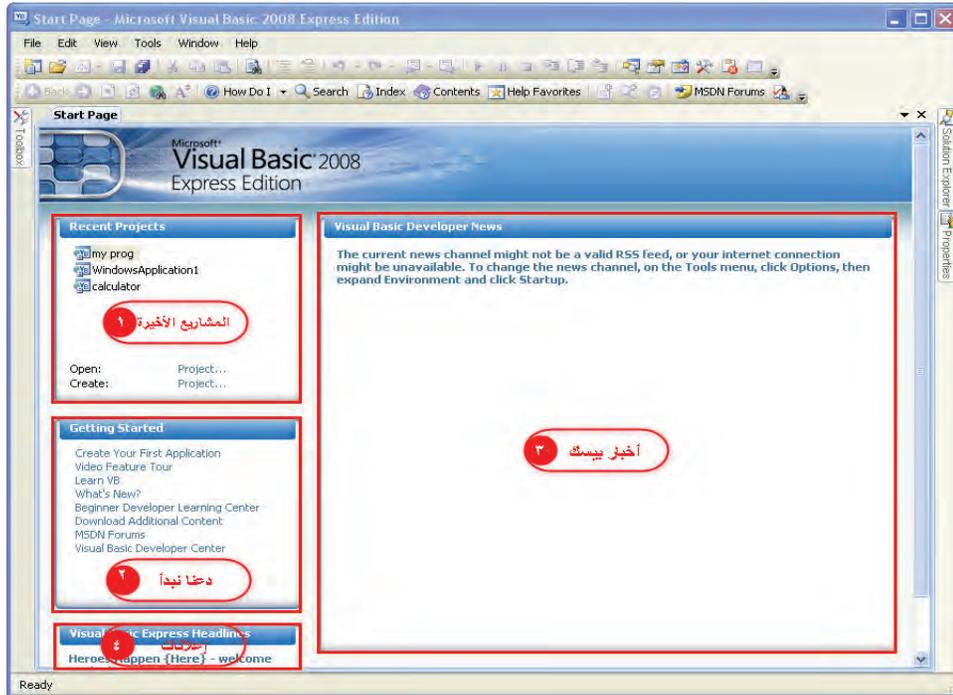
ثانياً : مزايا البيسك المرئي ٢٠٠٨

- (1) تعتمد على تعليمات سهلة الفهم
مثل تعليمة: **If ... then الشرط** وتعني إذا كان ... نفذ
- (2) تعتمد على كائنات رسومية كائنات رسومية تسهل الاتصال مع المستخدم
Button, CheckBox, CheckedListBox, ComboBox, DateTimePicker
- (3) لغة برمجة تستجيب للأحداث
أي يتم تنفيذ التعليمات البرمجية بعد أن يقوم مستخدم البرنامج بفعّل مثل الضغط على زر خروج
- (4) سهولة اكتشاف الأخطاء البرمجية
فمثلاً عند زيادة حرف d على تعليمة End يظهر خط متعرج أسفل الكلمة للتنبيه على وجود خطأ
Private Sub Button1_Click()
End() End Sub
- (5) بناء أنواع تطبيقات مختلفة
يستطيع بناء تطبيقات :
(أ) النوافذ .
(ب) الويب .
(ج) الموبايل .

ثالثاً: تشغيل البيسك المرئي ٢٠٠٨

لتشغيل لغة البيسك المرئي نتبع الخطوات الآتية :

اضغط على زر ابدأ ← كافة البرامج ← Microsoft Visual Basic 2008 Express Edition تظهر لنا النافذة الرئيسية محتوية على صفحة البدء Start Page مقسمة إلى أربعة أجزاء كما هو واضح في الشكل الآتي :



والجدول الآتي يوضح وظائف الأجزاء المشار إليها في الرسم :

الوظيفة	الجزء	الرقم
(أ) تعرض آخر مشاريع تم التعامل معها لفتحها بصورة سريعة . (ب) لفتح مشروع سبق إنشاؤه من خلال الخيار Open Project . (ج) لإنشاء مشروع جديد من خلال الخيار Create Project .	قائمة المشاريع الأخيرة Recent Projects	١
تعطي معلومات مبسطة عن استخدام البيسك المرئي للمستخدم المبتدئ.	دعنا نبدأ Getting Started	٢
وتظهر فيها أخبار لغة البيسك المرئي وتحديث تلقائياً أثناء الاتصال بالإنترنت .	الأخبار News	٣
تظهر روابط إنترنت لرؤوس مواضيع وإعلانات البيسك المرئي ، ولا بد من الاتصال بالإنترنت	إعلانات Headlines	٤

المشروع Project هو البرنامج أثناء مرحلة التصميم.

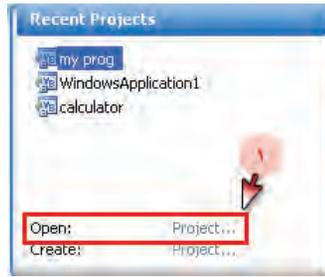
رابعاً: فتح مشروع سبق إنشاؤه

في الفصل التالي سوف نتعرف على واجهة البيسك المرئي Visual Basic 2008 وعناصرها، ولكي تتمكن من رؤية نوافذ الواجهة كاملة العناصر علينا أولاً أن نفتح مشروعاً تم إنشاؤه مسبقاً، ولفتح مشروع اتبع الخطوات الآتية:

إذا كان اسم المشروع يظهر في «قائمة المشاريع الأخيرة» يتم فتحه بالضغط على اسمه مباشرة دون الحاجة لتحديد موقعه.

أو اتبع الخطوات الآتية لفتح مشروع معلوم موقعه:

١ - اختر Open Project أسفل جزء المشاريع الأخيرة «Recent Projects».

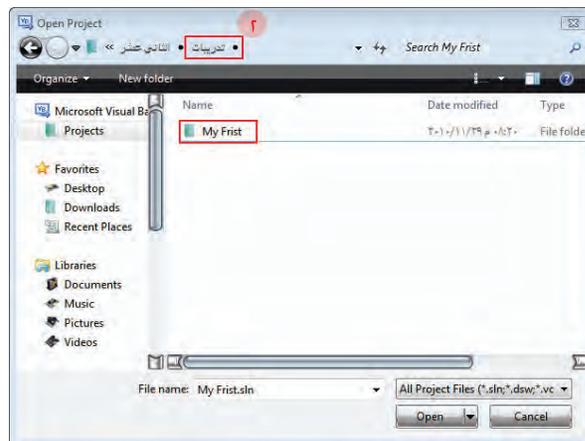


يمكن استخدام الأمر «Open Project» من قائمة «File» لفتح مشروع سبق إنشاؤه بنفس الخطوات السابقة.

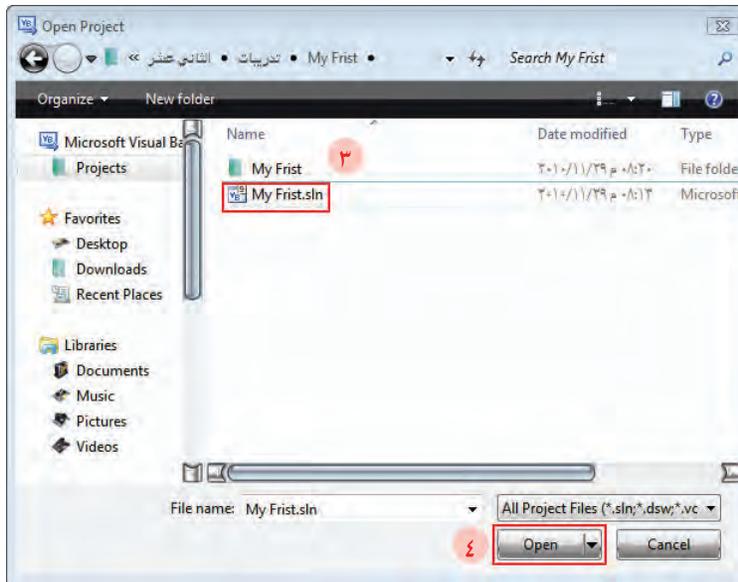


٢ - انتقل إلى مجلد «تدريبات» الذي يحتوي على المشروع، واضغط عليه ضغطاً مزدوجاً على مجلد

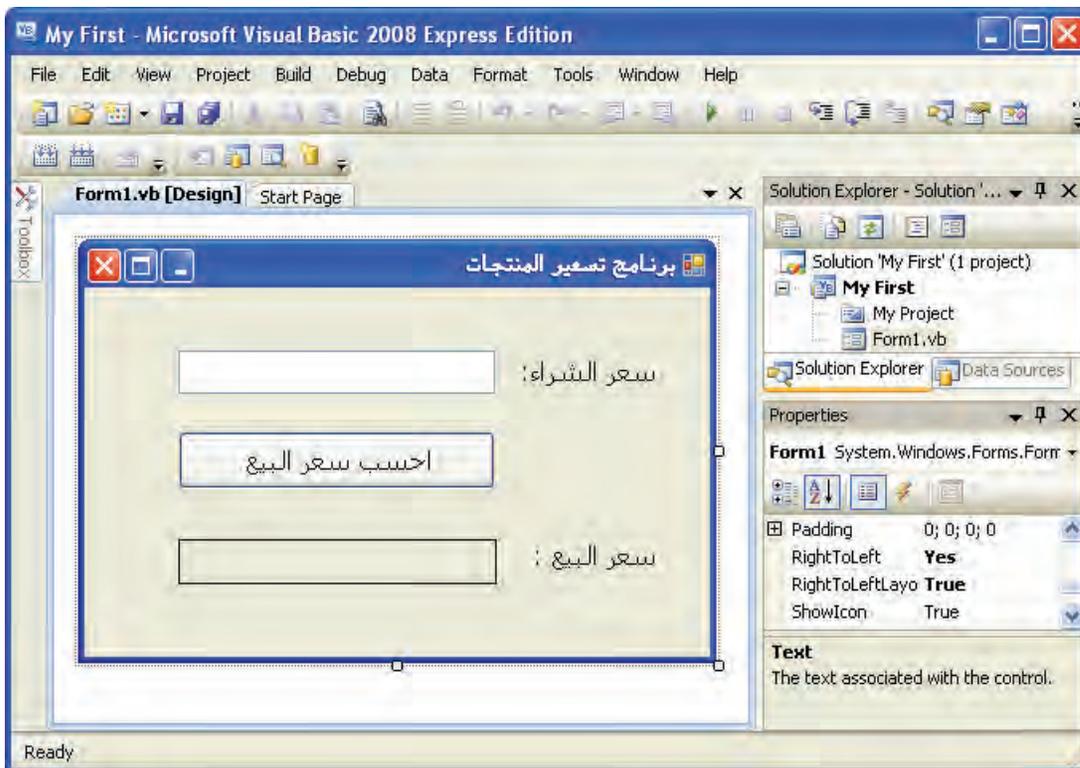
المشروع «My Frist».



٣ - اختر ملف المشروع المراد فتحه ، ثم اضغط زر «Open» .

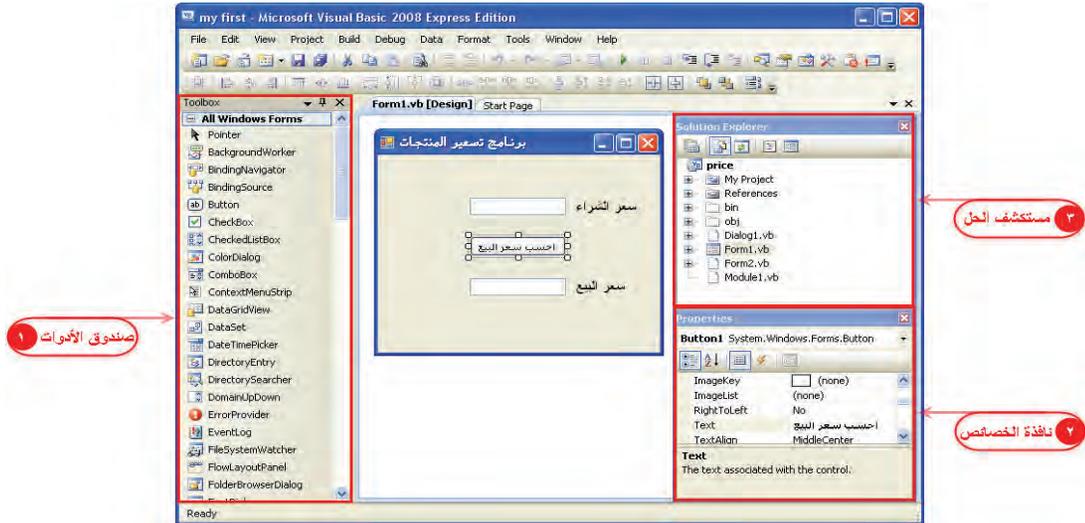


يتم فتح المشروع في صفحة تسمى (صفحة التصميم Design) وتظهر به نافذة النموذج Form محتوية على كائناتها.



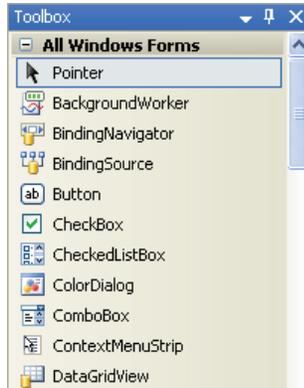
خامساً: عناصر واجهة البيسك المرئي ٢٠٠٨

بعد تثبيت ظهور العناصر الأساسية لواجهة البيسك المرئي ، تظهر على الشاشة ثلاث مناطق وهي :



(١) صندوق الأدوات Toolbox

هي نافذة يمكن من خلالها إضافة الكائنات المختلفة للنموذج ،



ويقسم صندوق الأدوات إلى مجموعات يمكن توسعه شجرتها من خلال الرمز (+) بجانبها، أو طيها من خلال الرمز (-)، وتحتوي كل مجموعة على العديد من الكائنات التي تتفق من حيث الوظيفة مثل كائنات الطباعة Printing، وكائنات صناديق الحوار Dialogs.

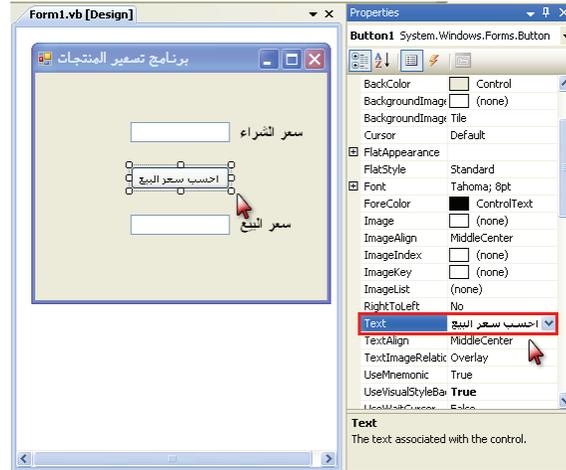
ومن أشهر المجموعات مجموعة (Common Controls) التي تحتوي على أكثر الكائنات استخداماً.

وفي حالة إغلاق صندوق الأدوات Toolbox نهائياً، يمكن إظهاره مرة أخرى من خلال الأمر «Toolbox» من قائمة «View» أو أداة  من شريط الأدوات.



(٢) نافذة الخصائص Properties

هي نافذة يمكن من خلالها ضبط أوصاف كائن (نصه أو نوع خطه أو طوله أو عرضه أو لونه... الخ). مثلاً الشكل الآتي يوضح قيمة خاصية «النص Text» للكائن المحدد (زر Button) والتي تساوي «احسب سعر البيع»،

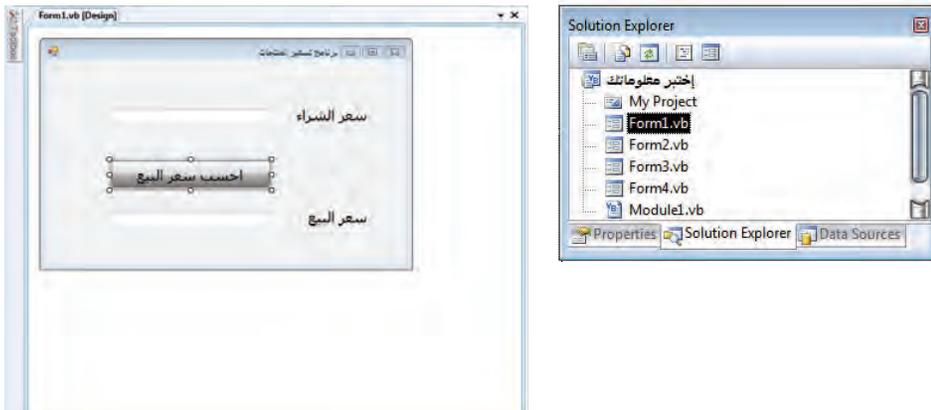


وفي حالة إغلاق نافذة الخصائص Properties نهائياً، يمكن إظهاره مرة أخرى من خلال الأمر «Properties Window» من قائمة «View» أو أداة  من شريط الأدوات.

تختلف محتويات نافذة الخصائص حسب الكائن المحدد .

(٣) نافذة مستكشف الحل Solution Explorer

هي نافذة تظهر بها جميع محتويات المشروع من النماذج Forms والوحدات النمطية Modules.

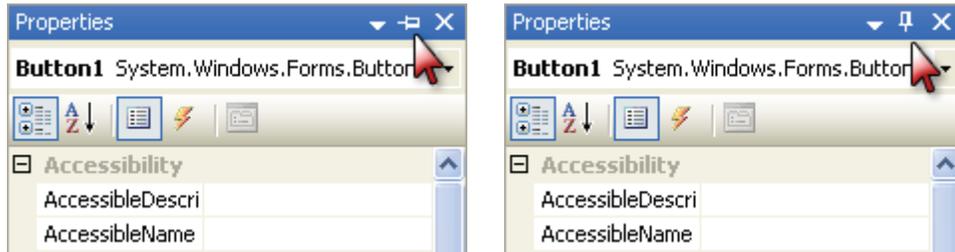


وفي حالة إغلاق نافذة مستكشف الحل Solution Explorer نهائياً، يمكن إظهاره مرة أخرى من خلال الأمر «Solution Explorer» من قائمة «View» أو أداة  من شريط الأدوات .



سادساً: التحكم في نوافذ الواجهة

جميع النوافذ التي تم ذكرها تتمتع بخاصية «الإخفاء التلقائي» «Auto Hide» أي تظهر عند مرور مؤشر الفأرة عليها .



ويمكن إبقاؤها ظاهرة بصورة دائمة بالضغط على الأداة  في شريط عنوانها، فيتغير شكل الأداة إلى الشكل  وللرجوع إلى وضع «الإخفاء التلقائي» نضغط على الأداة مرة أخرى .

في وضع الإظهار الدائم للنوافذ السابقة يمكن نقل النوافذ السابقة إلى أي مكان على الشاشة، وذلك بالسحب والإفلات بالفأرة من شريط عنوانها .



وللرجوع على الوضع الافتراضي نختار الأمر «Reset Windows layout» من قائمة Window .

لتفعيل خاصية «الإخفاء التلقائي» لجميع النوافذ السابقة نستخدم الأمر «Auto Hide All» من قائمة Window .





الباب الثاني

مراحل بناء البرنامج

الفصل الأول

بداية بناء برنامج

الفصل الثاني

ضبط عناصر البرنامج

الفصل الثالث

كتابة التعليمات البرمجية



visual Basic



الفصل الأول

بداية بناء البرنامج

التخطيط للبرنامج VB

إنشاء مشروع جديد VB

إضافة أدوات تحكم VB



visual Basic



تمر عملية بناء برنامج بعدة مراحل متتالية ومتراطة ، نوضحها في الشكل الآتي:



أولاً: التخطيط للبرنامج

للتخطيط لأي برنامج حاسوبي ، علينا تحديد ثلاثة أشياء مهمة نوضحها في الشكل الآتي :



- ١ - المدخلات : هي البيانات التي يجب أن تدخل للبرنامج للحصول على المخرجات .
- ٢ - المعالجة : هي العمليات البرمجية التي ستؤثر على البيانات .
- ٣ - المخرجات : هي المعلومات المطلوب الحصول عليها .

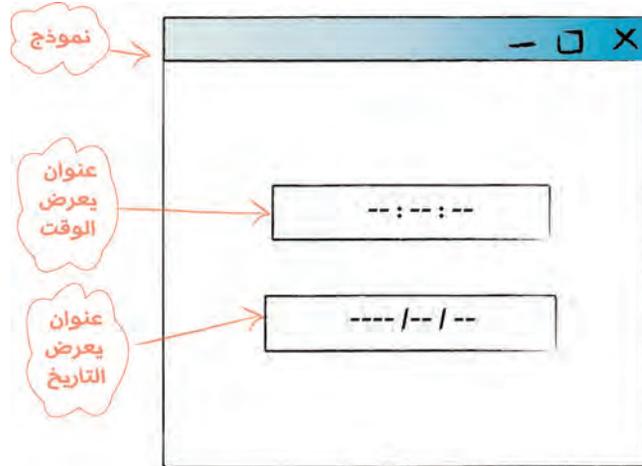
بعض البرامج لا تحتاج إلى مدخلات من المستخدم ، حيث تعتمد على مزايا داخل الجهاز أو تستقي بياناتها من الانترنت .



مثال :

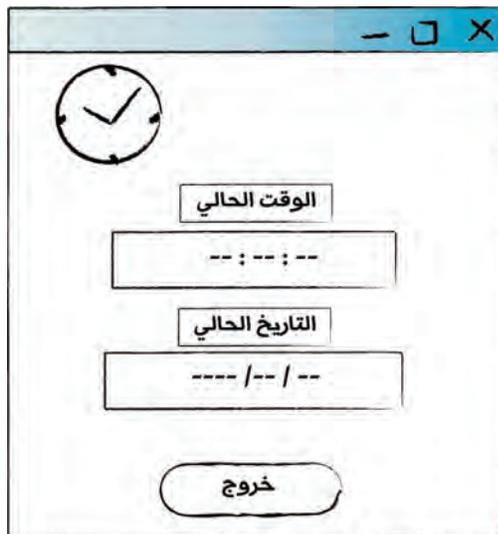
نود عمل برنامج يعرض الوقت والتاريخ الحالي، وهذا البرنامج لا يحتاج إدخال بيانات من المستخدم إنما يستقي بياناته من الساعة الداخلية بالحاسوب.

قبل الشروع في البرنامج نخطط على الورق كيف سيبدو شكل البرنامج :



وحتى يبدو البرنامج بشكل أجمل نضيف بعض الكائنات الثانوية التي تحسن من مظهر البرنامج:

- (أ) تسمية لعنوان الوقت .
- (ب) تسمية لعنوان التاريخ .
- (ج) صورة معبرة عن البرنامج .
- (د) زر للخروج من البرنامج .

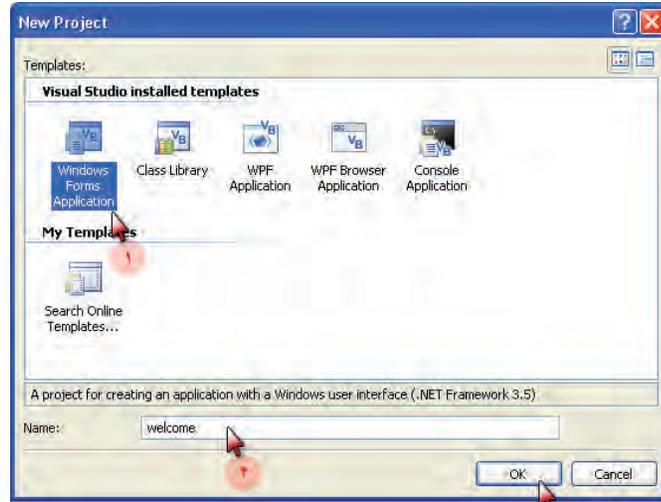


ثانياً: إنشاء مشروع جديد

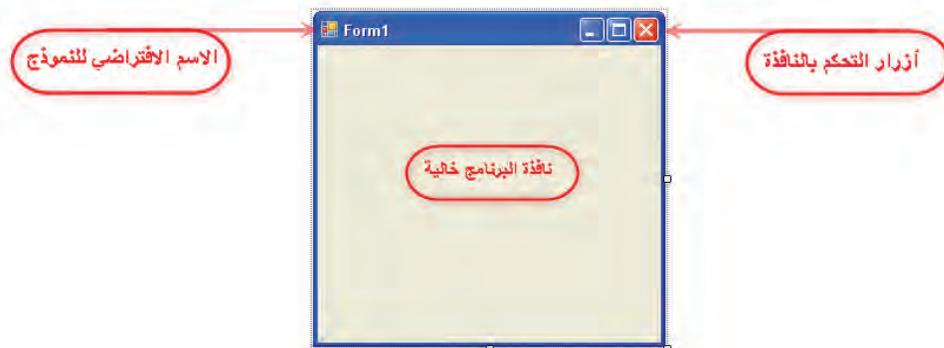
عند إنشاء مشروع جديد من خلال البيسك المرئي Visual Basic 2008، ينشئ البرنامج في البداية حاوية أكبر من مشروعك تسمى «الحل Solution» تتيح عمل أكثر من مشروع لموضوع واحد. وفيما يلي خطوات إنشاء مشروع جديد ضمن حل جديد:

(١) إنشاء حل Solution جديد

- ١ - شغل البيسك المرئي ٢٠٠٨.
- ٢ - اختر الأمر «New project» من قائمة «File»، أو الأداة  من شريط الأدوات.
- ٣ - من صندوق حوار «New project»، اختر القالب «Windows Forms Application».
- ٤ - في صندوق الاسم Name نستبدل الاسم الافتراضي للحل (Windows forms Application) بالاسم «Welcome»، ثم اضغط زر «Ok».



يتم إنشاء نموذج واجهة البرنامج كما في الشكل الآتي:



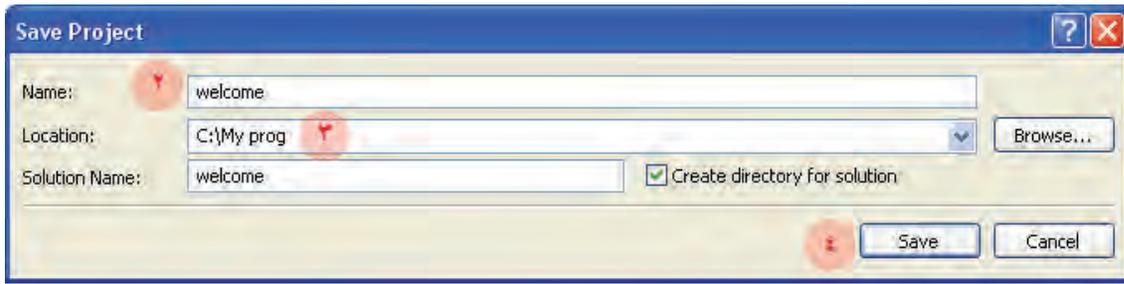
يظهر للنموذج الجديد العنوان الافتراضي "Form1" في شريط العنوان.
كما تظهر أزرار التحكم بالنافذة  تلقائياً في شريط العنوان.



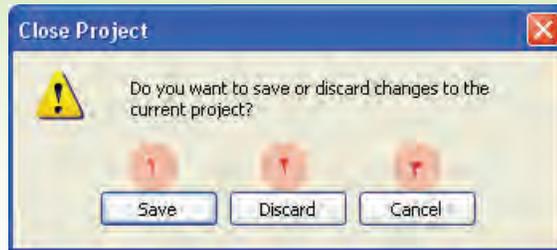
(٢) حفظ مشروع Project جديد :

في الخطوة السابقة أنشأت حلاً ولم يكتمل إنشاء المشروع لذلك يفضل حفظ المشروع في هذه المرحلة ولحفظ المشروع اتبع الآتي:

- ١ - اختر الأمر «Save All» من قائمة «File»، فيظهر صندوق حوار «Save Project».
- ٢ - يسمي البرنامج المشروع باسم الحل تلقائياً في خانة Name ويفضل تركه كما هو.
- ٣ - حدد المجلد الذي ستحفظ فيه مشروعك.
- ٤ - اضغط على الزر «Save».



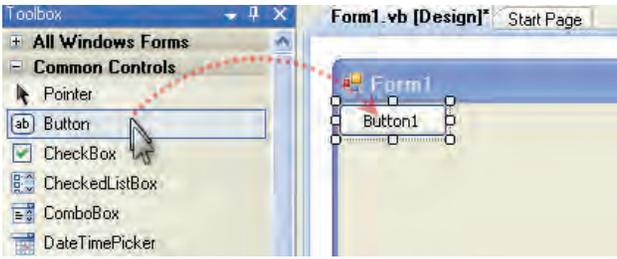
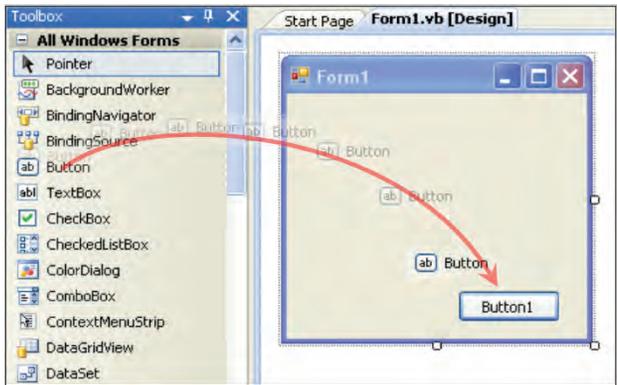
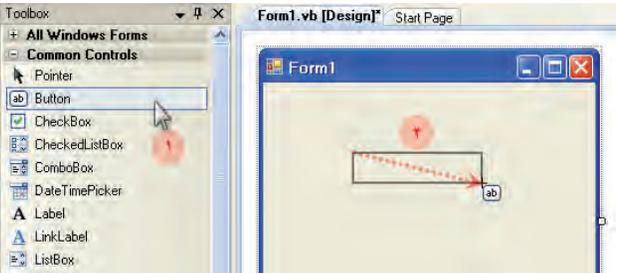
في حالة إغلاق بيئة البيسك المرئي أو إغلاق المشروع قبل حفظ التعديلات ، يظهر صندوق حوار «Close Project» الذي ينبهنا بأنه لم يتم حفظ التعديلات التي تمت :



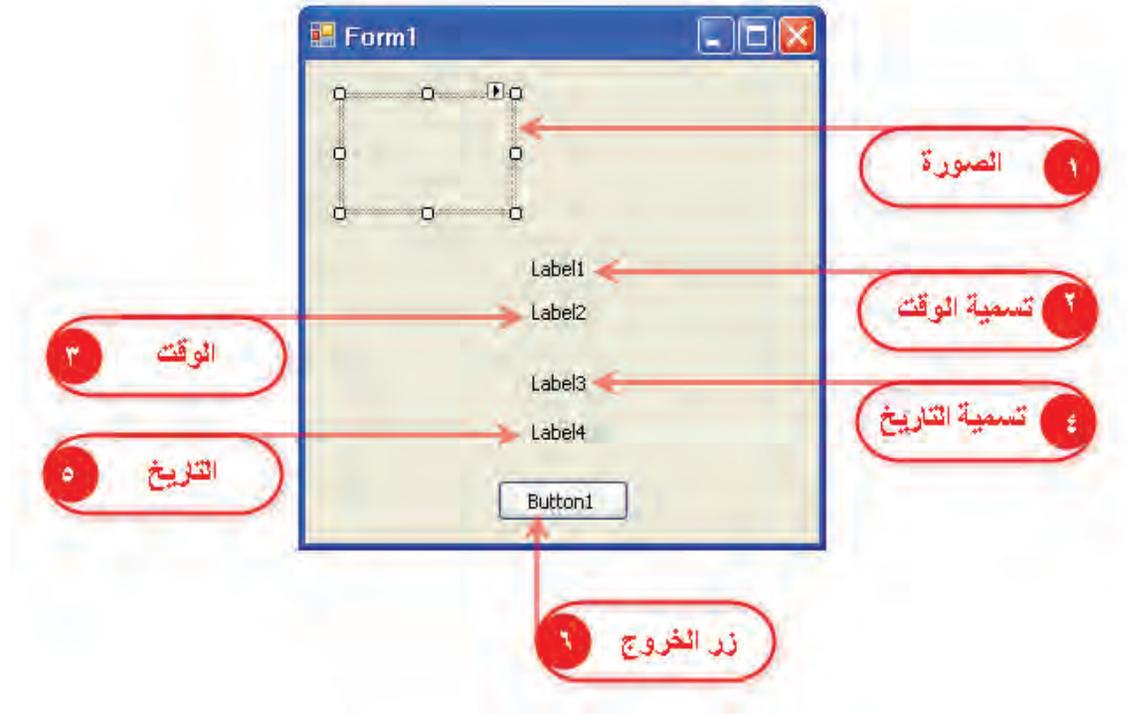
- حيث :
- (١) الزر «Save» : لإغلاق المشروع مع حفظ التعديلات .
 - (٢) الزر «Discard» : لإغلاق المشروع وعدم حفظ التعديلات .
 - (٣) الزر «Cancel» : لعدم إغلاق المشروع .

ثالثاً: إضافة أدوات التحكم

لإضافة أداة من صندوق الأدوات Toolbox إلى النموذج، يمكن استخدام إحدى الطرق الثلاث التي نوضحها في الجدول الآتي:

الطريقة	الوظيفة	صورة تعبر عن الطريقة
الضغط المزدوج على الأداة	يضيف العنصر بحجمه الافتراضي أعلى يسار النموذج في حالة تفعيله أو فوق العنصر الفعال حالياً.	
سحب الأداة وإفلاتها على النموذج	تتم إضافة العنصر بحجمه الافتراضي في المكان الذي تم فيه تحرير زر الفأرة.	
اختيار الأداة ثم رسم الأداة على النموذج	تتم إضافة العنصر بالحجم الذي تم رسمه والمكان الذي تم الرسم فيه.	

و بالعودة إلى المشروع الذي خططنا لإنشائه في بداية الفصل ، سوف نقوم بإضافة الأدوات إلى النموذج الجديد باحدى الطرق السابقة حتى يصبح النموذج بالصورة التالية :



مستعيناً بالأدوات التالية من شريط الأدوات :

الوظيفة	صورة الأداة	اسم الأداة
يظهر بداخله الصور	PictureBox	صندوق صورة
يظهر بداخله العناوين	Label	عنوان
يرتبط بتفعيل تعليمات برمجية عند الضغط عليه بالفأرة	Button	زر

الفصل الثاني

ضبط عناصر البرنامج

تعريف الخاصية VB

تغيير خاصية كائن أثناء
التصميم VB



visual Basic



لضبط عناصر البرنامج نستعين بنافذة الخصائص التي تظهر في الوضع الافتراضي يمين واجهة البيسك المرئي ٢٠٠٨ .

أولاً: تعريف الخاصية

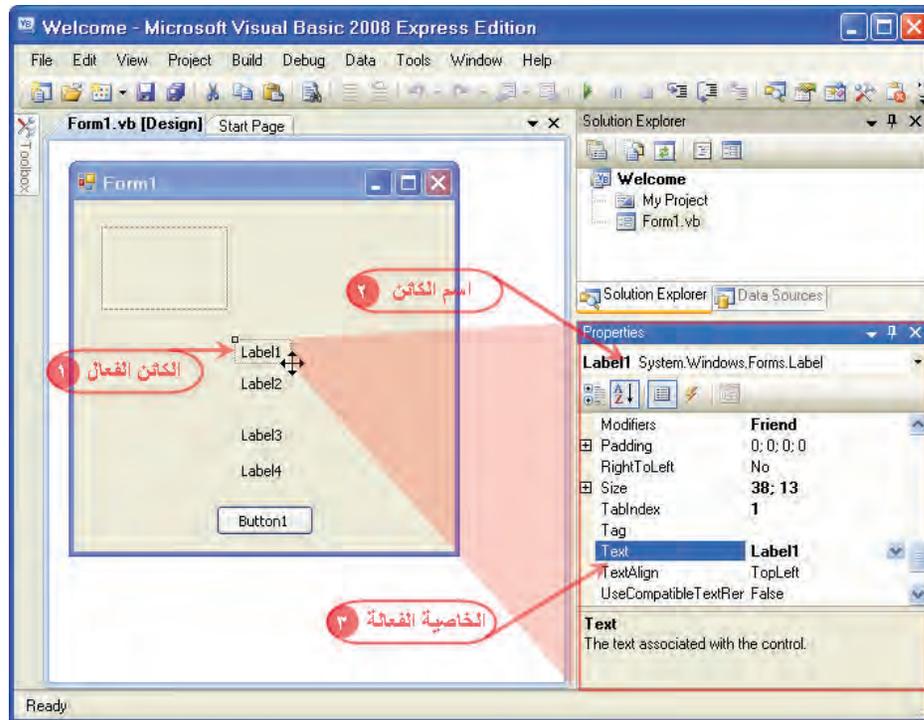
هي أحد أوصاف الكائن التي تحدد مظهره وسلوكه أثناء التصميم ووقت التشغيل.

يمكن تغيير خصائص الكائن أثناء التصميم من خلال نافذة الخصائص وأيضاً يمكن تغيير خاصية للكائن أثناء تشغيل البرنامج من خلال التعليمات البرمجية.

ثانياً: تغيير خاصية كائن أثناء التصميم

لتغيير خاصية لكائن علينا:

- ١ - تفعيله بالضغط عليه بزر الفأرة الأيسر.
- ٢ - نختار الخاصية المطلوبة من نافذة الخصائص ونقوم بتغييرها.



عند تحديد أي كائن يحدث ما يلي في نافذة الخصائص:

- ١ - يتغير اسم الكائن أعلى نافذة الخصائص حسب الكائن المحدد.
- ٢ - تتغير الخصائص إلى الخصائص المناسبة للكائن المختار.



وتختلف طريقة تحرير الخاصية في نافذة الخصائص كما في الجدول الآتي :

الشكل	مثال	الطريقة	اسم الطريقة
	خاصية محاذاة النص Text Align توجد داخل قائمة	الضغط على زر القائمة بجانب الخاصية ثم اختيار أحد قيمها.	الاختيار من قائمة
	تغيير خاصية الخط Font لكائن .	الضغط على زر صندوق خيارات الأداة (...) بجانب الخاصية المطلوبة.	فتح صندوق حوار
	تغيير خاصية النص Text لكائن .	الكتابة داخل صندوق الخاصية مباشرة بعد وضع مؤشر الكتابة به .	التحرير

بعض الخصائص لها خصائص فرعية مثل خاصية الحجم Size، ولإظهار هذه الخصائص الفرعية نضغط على رمز (+) بجانب خاصية Size، فتظهر الخاصيتان العرض Width والارتفاع Height .

⊕	MinimumSize	0; 0
⊕	Padding	0; 0; 0; 0
⊖	Size	100; 23
	Width	100
	Height	23

في الفصل السابق قمنا بالتخطيط لعمل برنامج يعرض الوقت والتاريخ الحالي، كما قمنا بإنشاء مشروع جديد، وأضفنا الكائنات إلى النموذج الرئيسي للمشروع، سوف نستكمل المشروع السابق بضبط خصائص كائنات المشروع أثناء التصميم.

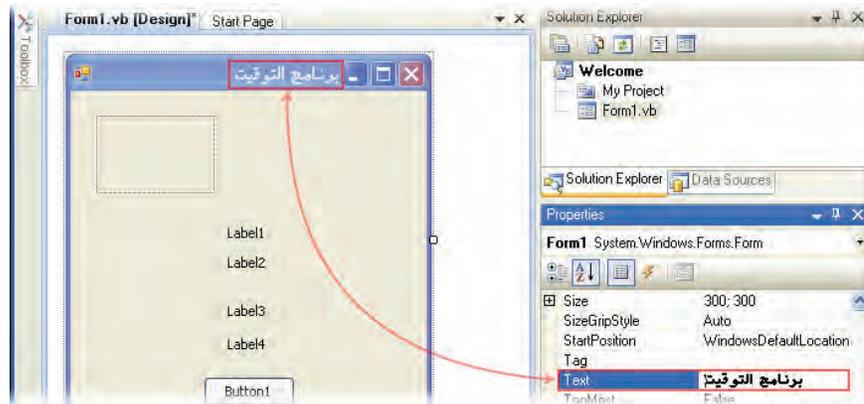
(١) خاصية الاسم Name :

تحدد هذه الخاصية اسماً فريداً للكائن في النموذج الحالي، ويتعامل مع هذا الاسم من خلال التعليمات البرمجية.

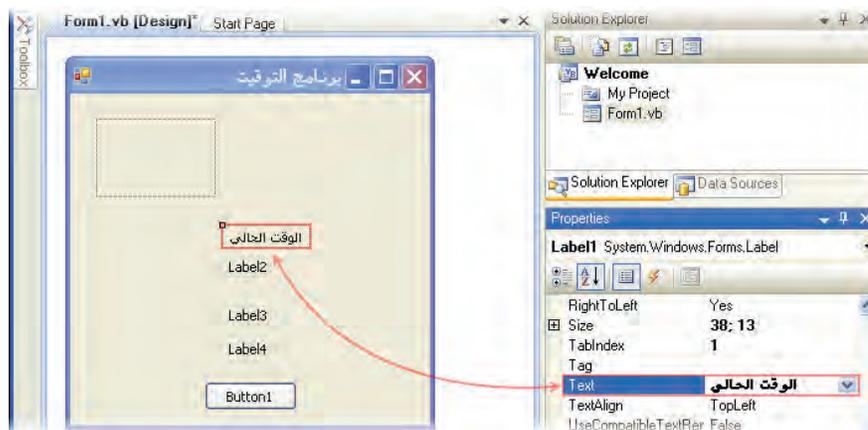
يسمي البيسك المرئي اسماً تلقائياً لكل كائن تضيفه للنموذج فمثلاً عند إضافة عنوان جديد يسميه Label1، وعند إضافة العنوان الثاني يسميه Label2... وهكذا.

(٢) خاصية النص Text :

تحدد هذه الخاصية النص الذي سيعرض داخل الكائن، وتحدد لكل كائن بشكل منفصل. في البداية حدد النموذج وفي الخانة المقابلة لخاصية النص Text اكتب «برنامج التوقيت»



وبالمثل حدد العنوان label1 واكتب في الخانة المقابلة لخاصية النص Text عبارة «الوقت الحالي» :



وبالمثل عنوان label3 نغير قيمة خاصية النص Text إلى «التاريخ الحالي»، وبالمثل زر Button1 نغير خاصية النص Text إلى «خروج» بحيث يصبح النموذج بهذه الصورة :

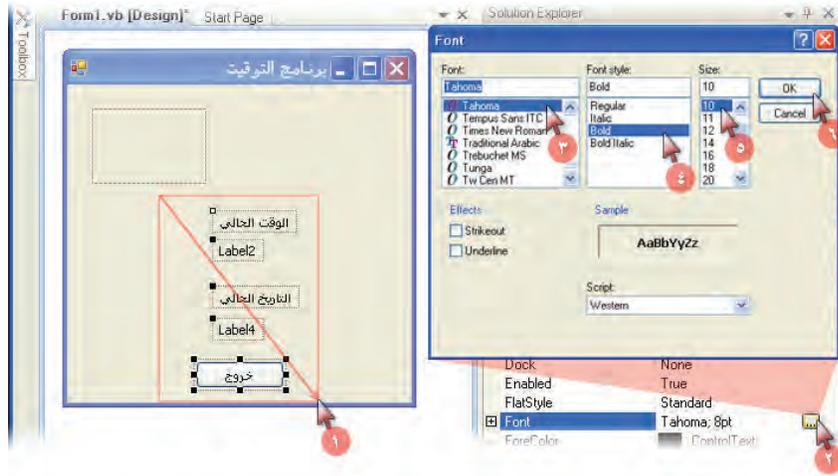


(٣) خاصية الخط Font :

هي خاصية تحدد نوع وحجم ونمط الخط للنص الذي يظهر داخل الكائن ، يمكن ضبطها لأكثر من كائن دفعة واحدة .

للاستفادة من هذه الخاصية في مشروعنا ، سنجعل العناوين والزر بخط أكبر وسميك كما يلي :

- ١ - نحدد كائنات العناوين والزر بالسحب بالفأرة أو بضغط مفتاح Shift مع الضغط على الكائنات بزر الفأرة الأيسر.
- ٢ - من نافذة خصائص الخاصية font نضغط على زر صندوق خيارات الخط.
- ٣ - نكبر حجم الخط إلى ١٠ ونختار النمط Bold ثم زر ok ، كما في الشكل التالي :



وبعيداً عن نافذة الخصائص يمكن الاستفادة من المزايا المتوفرة في قوائم البيسك المرئي ٢٠٠٨، مثل ميزة توسيط الكائنات بالنسبة لبعض من خلال أمر «توسيط Centers» المتفرع من أمر «محاذاة Align» من قائمة «تنسيق Format» :



كما يمكن عمل توسيط للكائنات بالنسبة للنموذج من خلال أمر «أفقي Horizontally»



(٤) خاصية الإرساء Anchor :

هي خاصية تحدد محاذاة الكائن عند تغير حجم النافذة أثناء التشغيل ، ويمكن ضبطها لأكثر من كائن دفعة واحدة .

للاستفادة من هذه الخاصية في مشروعنا ، بحيث نضمن أن الكائنات السابقة تظل في وسط النافذة مهما تغير حجمها أثناء التشغيل :

- ١ - نبقى تحديد الكائنات السابقة .
- ٢ - نضغط على زر القائمة المجاور لخاصية الإرساء Anchor .
- ٣ - نضغط ضغطتين على المحاذاة ليسار لإلغائه حتى تكون المحاذاة إلى أعلى فقط .



(٥) خاصية اللون الأمامي ForeColor :

تستخدم لتغيير لون الخط لنص الكائن المحدد ، ويمكن ضبطها لأكثر من كائن دفعة واحدة .
للاستفادة من هذه الخاصية في مشروعنا ، نغير لون خط العناوين التي يظهر بها الوقت والتاريخ
وذلك من خلال :

- ١- تحديد كل من عنوان Label2 والعنوان label4 معاً .
- ٢- نضغط على زر القائمة بجوار خاصية اللون الأمامي ForeColor .
- ٣- نختار بطاقة Custom .
- ٤- نختار اللون الأحمر Red .



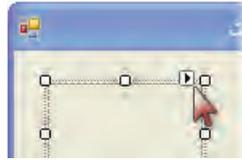
(٦) خاصية الصورة Image :

تحدد هذه الخاصية الصورة التي تظهر داخل الكائن ، وتحدد بشكل منفصل لكل كائن .
للاستفادة من هذه الخاصية في مشروعنا ، سنضع صورة ساعة داخل كائن الصورة Picture1
من خلال الخطوات التالية :

- ١- حدد كائن الصورة PictureBox1 .
- ٢- اضغط على زر صندوق الحوار بجوار خاصية الصورة Image .
- ٣- عند ظهور صندوق Select Resource ، اضغط على زر إحضار Import .
- ٤- حدد الصورة المطلوبة «Clock» .



يمكن استخدام زر السهم الذي يظهر عند تحديد كائن الصورة مباشرة في تحديد الصورة التي تظهر داخل كائن صورة .



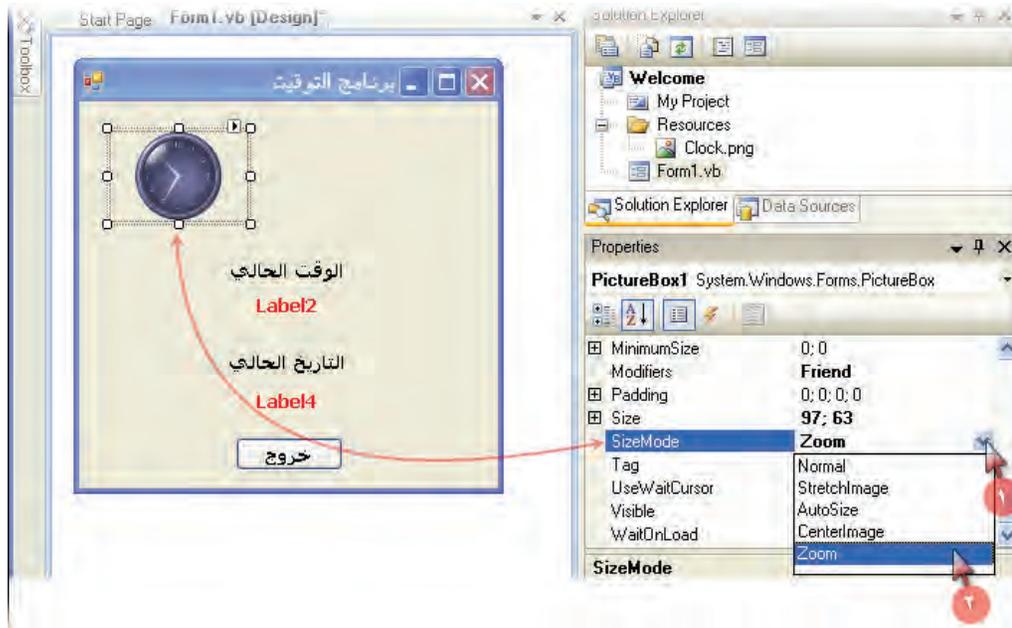
(٧) خاصية نمط الحجم SizeMode

تحدد هذه الخاصية طريقة عرض الصورة داخل إطار كائن ، وقيم هذه الخاصية يوضحها الجدول التالي :

الخيار	المعنى	الوظيفة
Normal	عادي	يظهر جزء من الصورة يناسب حجم الإطار.
StretchImage	تمدد الصورة	يجعل الصورة تتلاءم مع الإطار.
AutoSize	تحجيم	يغير حجم الإطار ليتلاءم مع الصورة.
Center	توسيط	يظهر جزء من وسط الصورة يناسب حجم الإطار.
Zoom	زووم	يجعل الصورة تساوي حجم الإطار مع المحافظة على تناسب الطول مع العرض.

وللاستفادة من هذه الخاصية في مشروعنا ، سنجعل الصورة تظهر بشكل كامل ومتناسق داخل كائن الصورة PictureBox كما يلي :

- ١ - حدد كائن الصورة PictureBox1 .
- ٢ - اضغط زر القائمة بجوار خاصية SizeMode .
- ٣ - اختر الخيار «زوم Zoom» .



الفصل الثالث

كتابة التعليمات البرمجية

تعريف التعليمات البرمجية VB

تعريف الحدث Event VB

محرر التعليمات البرمجية VB

اختبار البرنامج VB

بناء البرنامج VB



visual Basic



في الفصلين السابقين أضفنا كائنات البرنامج وضبطنا مظهرها ولكن حتى الآن يظل البرنامج بلا حياة وذلك لأننا إذا شغلنا البرنامج فلن يعرض الوقت الحالي ولا التاريخ الحالي ولن يعمل زر «خروج» حتى نقوم بإضافة التعليمات البرمجية التي ستجعل البرنامج يقوم بالمطلوب منه .

أولاً: تعريف التعليمة البرمجية

هي عبارات نصية اصطلاحية دقيقة تحدد للحاسوب عملاً محدداً يقوم به أثناء تشغيل البرنامج مثل End, DateString, TimeString

مثال :

تعليمة End تأمر الحاسوب بإغلاق البرنامج الحالي .
ومن أهم التعليمات البرمجية:

١ - تحديد قيمة لإحدى خصائص الكائن أثناء تشغيل البرنامج وتتبع القاعدة التالية:

ObjectName . Properties = Value
القيمة الخاصية اسم الكائن

مثال :

لتغيير لون خلفية النموذج الحالي للون السماوي أثناء تشغيل البرنامج نكتب التعليمة التالية :

Me . BackColor = Color.Aqua

حيث :

- (أ) Me تدل على النموذج الحالي مهما كان اسمه .
- (ب) BackColor خاصية لون الخلفية .
- (ج) Color.Aqua قيمة اللون السماوي .

لا يجوز حدوث أي خطأ إملائي في كتابة التعليمات البرمجية، ولكن لا تقلق عزيزي المتعلم فسوف يساعدك البيسك المرئي في كتابة التعليمة البرمجية بشكل سهل وبسيط كما سترى أثناء كتابة تعليمات في محرر التعليمات البرمجية لاحقاً .



٢ - تفعيل إحدى طرق الكائن Object Methods والطريقة هي إحدى الوظائف المضمنة في الكائن وتتبع القاعدة التالية:

ObjectName . Method
الطريقة اسم الكائن

مثال :

لإخفاء كائن الصورة أثناء تشغيل البرنامج نكتب التعليمية التالية :

`PictureBox1.Hide ()`

حيث :

- `PictureBox1` هو اسم الكائن المراد إخفاؤه أثناء التشغيل .
- `Hide ()` الطريقة المراد تنفيذها .

عكس طريقة الإخفاء `Hide ()` طريقة الإظهار `Show ()`



ثانياً : تعريف الحدث Event

هو الإجراء المطلوب تنفيذه في زمن محدد اعتماداً على التعليمية البرمجية.

مثال :

كتابة تعليمة `End` لكائن زر «الخروج» في حدث `Click` .

ويعني ذلك أنه عندما يضغط مستخدم البرنامج زر الفأرة الأيسر على زر «خروج» تنفذ تعليمة `End` وينتهي البرنامج .

وبذلك يرتبط بالكائن ثلاثة أشياء :

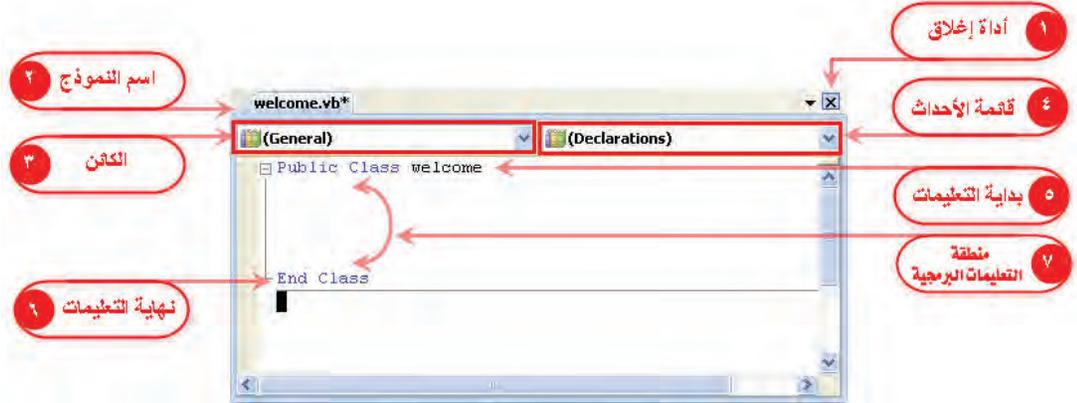
خصائص	Properties	تحدد مظهره
طرق	Method	وظائف مضمنة في الكائن
أحداث	Events	تؤثر على الكائن مثل الضغط بزر الفأرة أو الضغط المزدوج ... الخ .

ثالثاً : محرر التعليمات البرمجية

هو نافذة ذكية لكتابة التعليمات البرمجية بصورة سهلة .

لاستدعاء محرر التعليمات البرمجية نختار أمر "Code" من قائمة "View" أو نضغط على المفتاح F7 .

تظهر نافذة المحرر كما في الشكل الآتي :



م	اسم المنطقة	الوظيفة
١	أداة إغلاق	لإغلاق نافذة محرر التعليمات البرمجية .
٢	اسم النموذج	يظهر اسم النموذج متبوعاً بعلامة vb* للدلالة على أن هذة النافذة هي نافذة التعليمات البرمجية للنموذج الحالي .
٣	الكائن	يظهر اسم الكائن الذي يتم كتابة التعليمات له .
٤	قائمة الأحداث	تظهر قائمة بالأحداث الممكنة للكائن ، مثل الضغط على زر الفأرة أو مرور الفأرة على عنصر .
٥	بداية التعليمات البرمجية للنموذج	السطر الأول من جميع التعليمات البرمجية للنموذج وكائناته .
٦	نهاية التعليمات البرمجية للنموذج	السطر الأخير من جميع التعليمات البرمجية للنموذج وكائناته .
٧	منطقة التعليمات البرمجية	منطقة كتابة التعليمات البرمجية .

تعليمات برمجية مستخدمة للاستفادة من المفاهيم السابقة:

يستخدم للحصول على الوقت الحالي من الحاسوب	TimeString
يستخدم للحصول على التاريخ الحالي من الحاسوب	DateString
يستخدم لإنهاء البرنامج	End

للاستفادة من المفاهيم السابقة في مشروعنا، فإننا بحاجة إلى كتابة تعليمات برمجية تحقق ما يلي أثناء تشغيل البرنامج:

١ - تغيير محتوى العنوان Label2 بحيث يعرض الوقت الحالي من خلال التعليمة البرمجية:

`Label2.Text = TimeString`

حيث تقوم هذه التعليمة بتغيير خاصية نص Label2 إلى الوقت الحالي التي تنتجها القيمة `TimeString`.

٢ - تغيير محتوى العنوان Label4 بحيث يعرض التاريخ الحالي من خلال التعليمة البرمجية:

`Label4.Text = DateString`

حيث تقوم هذه التعليمة بتغيير خاصية نص Label4 إلى التاريخ الحالي الذي تنتجها القيمة `DateString`.

٣ - كتابة تعليمة إنهاء البرنامج مرتبطة بالزر Button1 من خلال التعليمة البرمجية:

`End`

ولكن يجب أن نختار الحدث المناسب لكتابة هذه التعليمات، كما يلي:

١ - بالنسبة لزر الخروج Button1 سيكون:

حدث Click أي الضغط بالزر الأيسر بالفأر على زر الخروج .

٢ - بالنسبة للعناوين سيكون :

حدث Load أي عند بدء تحميل النموذج .

يوفر البيسك المرئي ميزة الحدث التلقائي فبالنسبة:

١ - للنموذج Form الحدث التلقائي Load عند بدء تشغيل النموذج .

٢ - للزر Button الحدث التلقائي Click عند الضغط على الفأرة .

وعندما تضغط ضغطاً مزدوجاً على أي كائن فإنه ينقلك إلى نافذة محرر التعليمات البرمجية

ويكتب تلقائياً بداية إجراء فرعي ونهايته الذي يناسب الكائن والحدث التلقائي ،

ولتطبيق ذلك على مشروعنا نضغط ضغطاً مزدوجاً على زر Button1 :

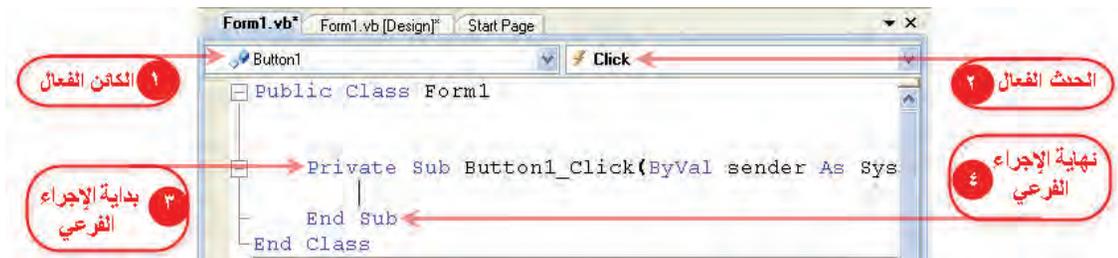


فتظهر لنا نافذة محرر التعليمات البرمجية وتلقائياً :

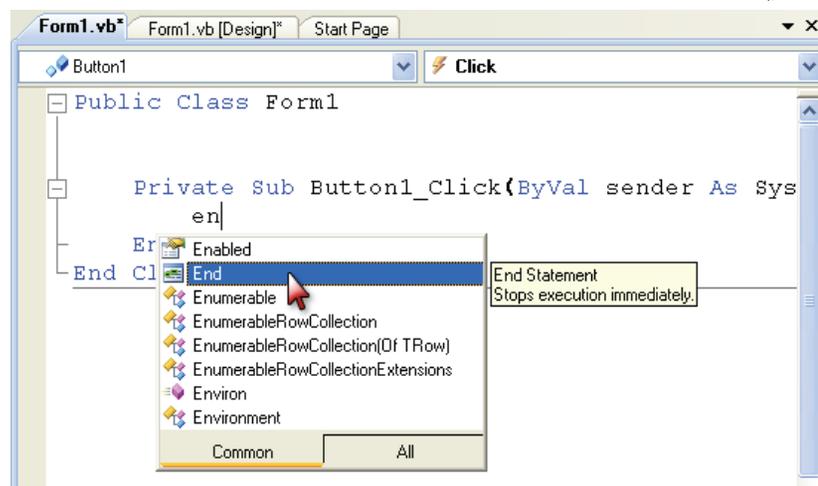
١ - الكائن الفعال : هو Button1.

٢ - الحدث الفعال : هو Click.

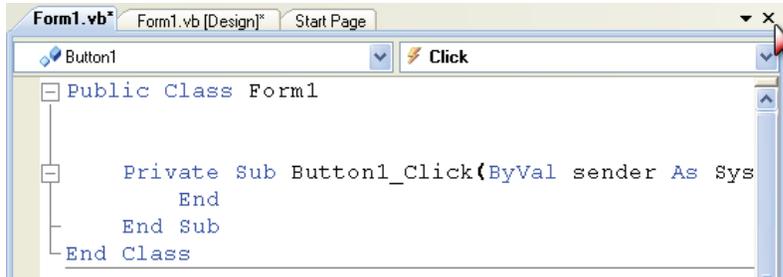
٣ - بداية ونهاية إجراء فرعي جديد باسم Button1_Click



بعد بداية الإجراء الجديد بمجرد كتابة أول حرفين من تعليمة End تظهر قائمة فرعية عند مؤشر الكتابة تساعدك في إكمال التعليمية ، اختر منها End.



وبعد ظهور التعليمة نغلق نافذة محرر التعليمات البرمجية:



```
Form1.vb* Form1.vb [Design]* Start Page
Button1 Click
Public Class Form1
    Private Sub Button1_Click(ByVal sender As Sys
        End
    End Sub
End Class
```

وبنفس الطريقة نضغط ضغطاً مزدوجاً على أي مكان من النموذج Form1 ، فتظهر لنا نافذة محرر التعليمات البرمجية وتلقائياً:

١- الكائن الفعال : هو Form1 .

٢- الحدث الفعال : هو Load .

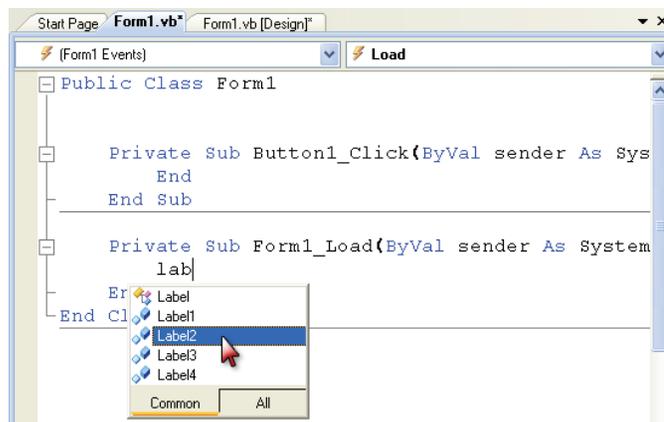
٣- بداية ونهاية إجراء فرعي جديد باسم Form1_Load



```
Start Page Form1.vb* Form1.vb [Design]*
(Form1 Events) Load
Public Class Form1
    Private Sub Button1_Click(ByVal sender As Sys
        End
    End Sub
    Private Sub Form1_Load(ByVal sender As System
        End Sub
End Class
```

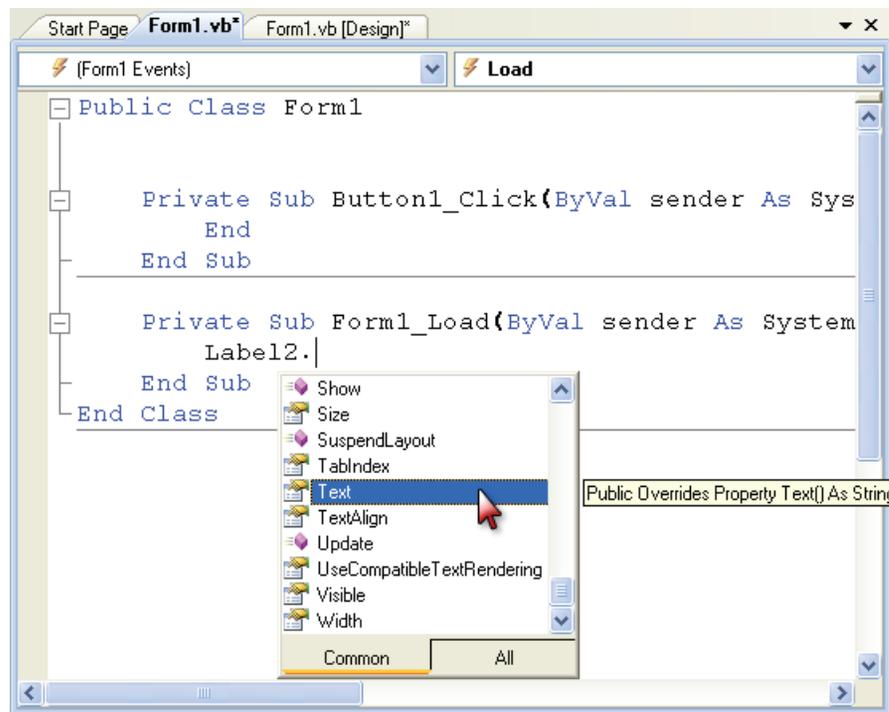
إجراء بدأ تشغيل النموذج

بعد بداية الإجراء الجديد نبدأ بكتابة الحرفين La فتظهر القائمة الفرعية عند مؤشر الكتابة فنختار منها Label2 .

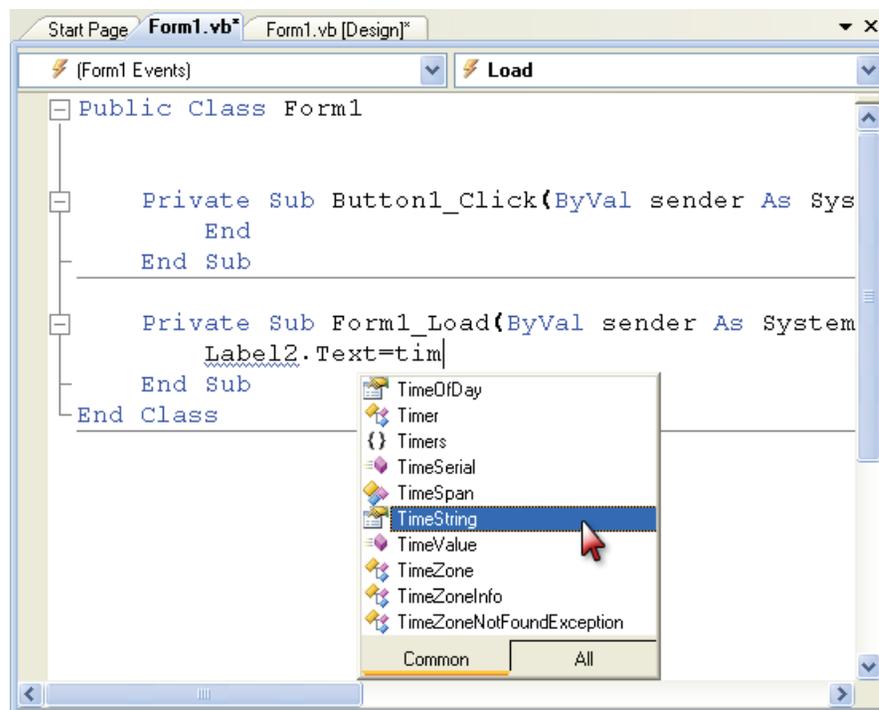


```
Start Page Form1.vb* Form1.vb [Design]*
(Form1 Events) Load
Public Class Form1
    Private Sub Button1_Click(ByVal sender As Sys
        End
    End Sub
    Private Sub Form1_Load(ByVal sender As System
        lab
    End Cl
    Label1
    Label2
    Label3
    Label4
Common All
```

نضع نقطة (.) ثم نكتب حرف T فتظهر القائمة الفرعية فنختار منها خاصية Text .



نكتب علامة التساوي ثم حرف T فتظهر قائمة القيم فنختار منها TimeString .



وبنفس الطريقة اكتب التعليمة البرمجية لتغيير العنوان Label4 ليعرض التاريخ كما في النافذة التالية:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As Sys
        End
    End Sub
    Private Sub Form1_Load(ByVal sender As System
        Label2.Text = TimeString
        Label4.Text = DateString
    End Sub
End Class
```

وبعد ذلك اختبر البرنامج من خلال الضغط على مفتاح F5 فيعمل البرنامج ويظهر الوقت والتاريخ الحالي:

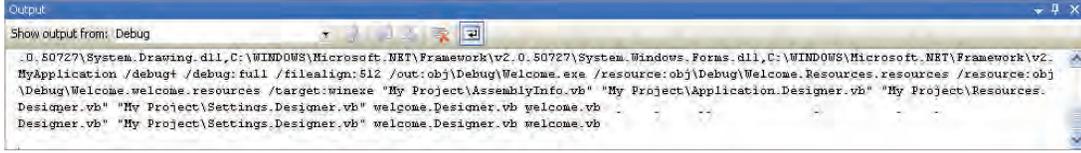


رابعاً: اختبار البرنامج

ولاختبار البرنامج والتأكد من صحة التعليمات البرمجية نستخدم إحدى الطرق التالية:

- ١ - أمر «إبدأ التدقيق Start Debugging» من قائمة «تدقيق Debug».
- ٢ - مفتاح الوظيفي F5.
- ٣ - أداة ▶ من شريط الأدوات.

عند اختبار البرنامج تظهر نافذة «Output» التي تظهر عملية اختبار التعليمات البرمجية ، كما في الشكل الآتي :



في حالة وجود أخطاء برمجية تظهر نافذة تحذيرية تفيد بوجود خطأ برمجي ، وتظهر الأخطاء داخل نافذة «Output» .

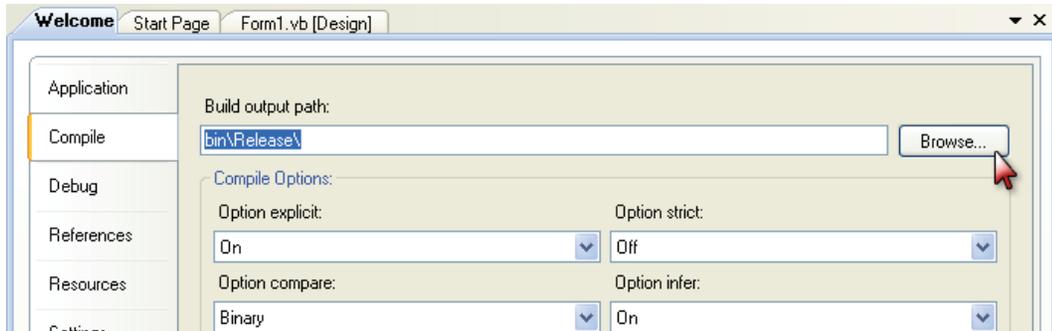


خامساً : بناء البرنامج

يحتاج المبرمج أن يقدم برنامجاً الذي أنشأه للمستخدمين في صورة ملف تنفيذي مستقل عن البيسك المرئي وتسمى هذه العملية بـ «بناء البرنامج» ،

قبل بناء البرنامج يفضل تحديد المكان الذي سيبنى فيه الملف التنفيذي للبرنامج ، حيث يضع البيسك المرئي بشكل تلقائي الملف التنفيذي للبرنامج في مجلد فرعي من مجلد المشروع هو bin/Release ، ولتغيير مكان بناء الملف التنفيذي للبرنامج اتبع الخطوات الآتية :

- ١- من قائمة «Project» اختر الأمر «Welcome Properties» حيث Welcome هو اسم المشروع.
- ٢- اختر Compile من الجزء الأيسر.
- ٣- ثم من خلال الزر «Browse» حدّد موقع المجلد الذي ترغب ببناء الملف التنفيذي للبرنامج فيه.





الباب الثالث

التعليمات البرمجية الأساسية

الفصل الأول

المتغيرات والثوابت

الفصل الثاني

التفرع



visual Basic



الفصل الأول

المتغيرات والثوابت

Variables المتغيرات VB

Constants الثوابت VB



visual Basic



تحتاج معظم البرامج أثناء عملها إلى بيانات متغيرة ومؤقتة تخصص لها مكان في ذاكرة الحاسوب ثم تمحوها مع إنهاء تشغيل البرنامج .

أولاً: المتغيرات Variables

(١) تعريف المتغير Variable

المتغير هو مكان في الذاكرة تخزن فيه بيانات أثناء تنفيذ البرنامج ، له اسم فريد يميزه ، ويختلف حجمه حسب نوع البيانات التي ستخزن فيه .

(٢) أنواع المتغيرات Variables Types

يتضمن البيسك المرئي ٢٠٠٨ أنواع متغيرات عديدة نلخص أشهرها في الجدول التالي :

نوع المتغير	نوع البيانات	يستخدم لـ	مثال	القيمة الافتراضية
Boolean	منطقي	تخزين البيانات التي لها احتمالان «نعم True» أو «لا False» .	إقامة	False
String	حرفي	تخزين البيانات النصية.	الاسم ، العنوان ، التقدير .	فارغ Null
Date	تاريخ /وقت	تخزين التاريخ أو الوقت أو التاريخ والوقت معاً.	تاريخ الميلاد ، تاريخ الالتحاق بالجامعة ، موعد الطائرة .	٠٠٠١/١/١
Byte	عددية	تخزين أعداد موجبة بدون فاصلة عشرية من ٠ إلى ٢٥٥ .	سنوات الخبرة ، العمر ، عدد أفراد الأسرة .	صفر
Integer	عددية	تخزين أعداد موجبة أو سالبة من - ٢١٤٧٤٨٣٦٤٧ إلى ٢١٤٧٤٨٣٦٤٨ بدون فاصلة عشرية .	عدد الطلاب ، عدد السلع .	صفر
Double	عددية	تخزين أرقام موجبه أو سالبة كبيرة ذات فاصلة عشرية .	حساب البنوك ، الأرقام الفلكية	صفر

يقصد بالقيمة الافتراضية القيمة التي تخصص للمتغير فور إنشائه ولم يطرأ عليها أي تعديل .



الإعلان عن المتغيرات Declaration of Variables (٣)

قبل استخدام أي متغير ضمن البرنامج يفضل الإعلان عنه داخل نافذة التعليمات البرمجية، والإعلان يعني إنشاء متغير جديد وتحديد اسم ونوع البيانات التي تخزن فيه، ويتم الإعلان عن المتغيرات باستخدام تعليمة Dim وهي تخضع للقاعدة التالية :

Dim Name As Type

حيث :

Name : اسم المتغير

Type : نوع المتغير

مثال :

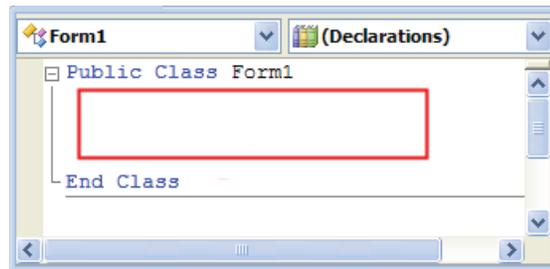


الكلمات Dim ، as ، كلمات مفتاحية تكتب أثناء الإعلان عن المتغير كما هي ، ولا يمكن استخدامها كأسماء للمتغيرات ، ليس هناك فرق بين كتابة أحرف كبيرة أو صغيرة في أسماء المتغيرات .

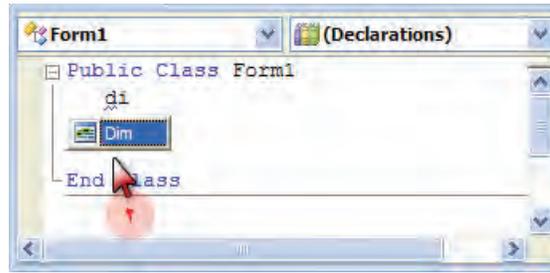


وفيما يلي خطوات الإعلان عن المتغيرات :

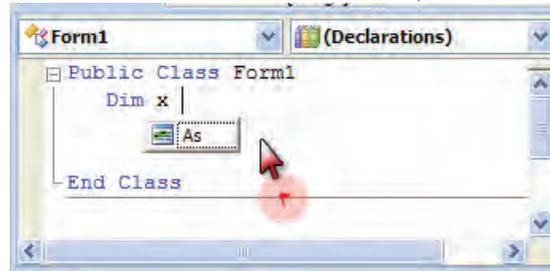
١- من خلال نافذة التعليمات البرمجية (Code) .



٢- نكتب الكلمة المفتاحية Dim ثم نترك مسافة .

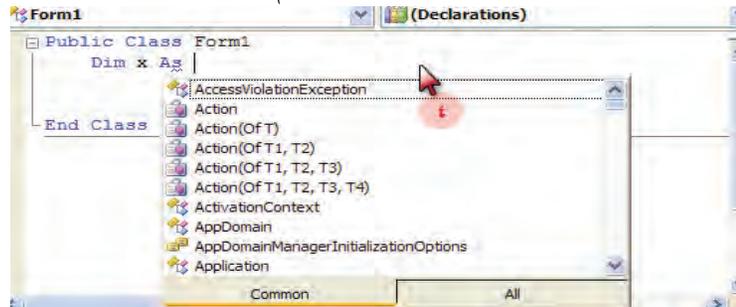


٣- نكتب اسم المتغير وليكن X ثم نترك مسافة.



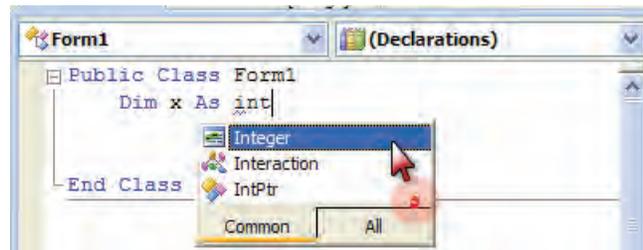
نلاحظ ظهور كلمة as بعد ترك مسافة بعد اسم المتغير.

٤- نكتب الكلمة المفتاحية as أو نختارها من القائمة ثم نترك مسافة.



نلاحظ ظهور قائمة بأنواع المتغيرات المتاحة في البرنامج.

٥- نكتب الأحرف الأولى من نوع المتغير وليكن Integer نلاحظ تصفية القائمة وظهور الأنواع التي تبدأ بالأحرف.



بعد الانتهاء من تنفيذ الخطوات السابقة يكون قد تم حجز مكان في الذاكرة له الاسم X وهو قادر على تخزين بيانات من نوع Integer.



يمكن التعبير عن أكثر من متغير ضمن عبارة Dim واحدة كما يلي :
أ- للإعلان عن متغيرات لها أنواع مختلفة تكتب كما يلي .

Dim x as Integer , y as single , z as date

ب- للإعلان عن متغيرات لها نفس النوع تكتب كما يلي .

Dim a,b,c as string

(٤) تخصيص قيم للمتغيرات

بعد الإعلان عن المتغيرات نحتاج لتخصيص قيم لها، ويعتبر معامل التساوي (=) هو المعامل الأساسي في عملية التخصيص، حيث يقوم بتخصيص القيمة الموجودة في جهة اليمين لمعامل التساوي إلى المتغير الموجود في جهة اليسار.

أمثلة على المعامل (=):

X = 5

تخصص القيمة 5 إلى المتغير X

Y = 6

تخصص القيمة 6 إلى المتغير Y

Z = X + Y

تخصص القيمة 11 إلى المتغير Z

يمكن تخصيص القيم للمتغيرات بطريقتين:

١- تخصيص قيم للمتغيرات أثناء تشغيل البرنامج.

٢- تخصيص قيم للمتغيرات أثناء كتابة التعليمات البرمجية (وقت التصميم).

مثال على تخصيص القيم للمتغيرات بالطريقتين:

The screenshot shows the Visual Basic IDE on the left and a runtime window titled 'حساب الزكاة' (Zakat Calculation) on the right. The code in the IDE is as follows:

```

Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim zakaa, money, total As Double
        zakaa = 0.025
        money = TextBox1.Text
        total = zakaa * money
        Me.Label4.Text = total
    End Sub
End Class
    
```

The runtime window shows a 'حساب الزكاة' window with a 'إجمالي المبلغ:' label and a text box containing '1000'. Below it, a 'الزكاة المستحقة:' label and a text box containing '20'. There are 'إغلاق' (Close) and 'الزكاة' (Zakat) buttons at the bottom.

لتخصيص القيم أثناء كتابة التعليمات البرمجية يراعى ما يلي :

(أ) عند إسناد قيم عددية تكتب الأرقام بدون علامات : $A=10$

(ب) عند إسناد قيم حرفية تكتب بين علامات تنصيص : $A= \text{“Ali”}$

(ج) عند إسناد تاريخ تكتب بين علامات ## : $Date= \# 25 / 11 / 1966 \#$



(٥) نطاق المتغيرات Variables Scope

أ - تعريف نطاق المتغير

هو الفترة الزمنية التي يظل فيها المتغير متاح أثناء تنفيذ البرنامج.

ب - أقسام المتغيرات وفقاً لنطاقاتها:

١- متغير على مستوى الإجراء الفرعي:

يتم الإعلان عن المتغير داخل إجراء فرعي، ويتم التعامل مع هذا المتغير داخل هذا الإجراء فقط.

مثال:

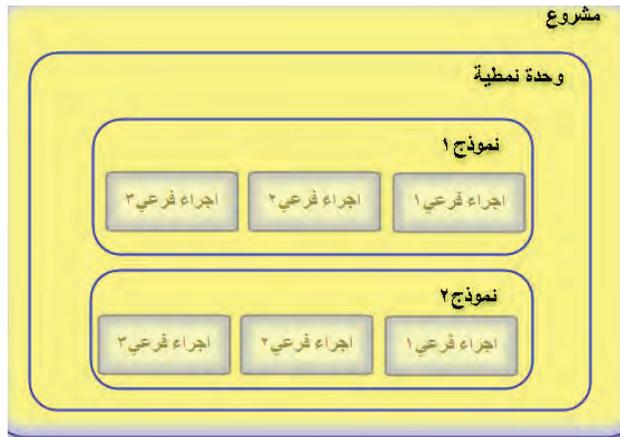
في المثال السابق «حساب الزكاة» كان كل من $Zakat$, $Money$, $total$ كلها متغيرات على مستوى الإجراء الفرعي الضغط على زر $Button1$ ، حيث خارج هذا الإجراء لا يعمل وإذا كتب في أي تعليمة برمجية سيعتبره برنامج البيسك المرئي خطأ.

٢- متغير على مستوى النموذج:

المتغير الذي يعلن داخل النموذج وقبل أي إجراء فرعي يكون نطاقه النموذج كله ويمكن تخصيص قيم له من أي مكان داخل النموذج.

٣- متغير على مستوى المشروع:

هو متغير نطاقه جميع نماذج المشروع، ويعلن عنه باستخدام التعليمة $Public$ وذلك ضمن وحدة نمطية $Module$ ويمكن تخصيص القيم له من أي نموذج داخل المشروع.



ج - تعريف الوحدة النمطية Module

هي ملف على صورة نافذة التعليمات البرمجية يتم الإعلان فيه عن متغيرات أو إجراءات يمكن الاستفادة منها في جميع نماذج المشروع .
وفيما يلي مثال تطبيقي يوضح كيفية بناء برنامج يعتمد على متغير على مستوى المشروع .

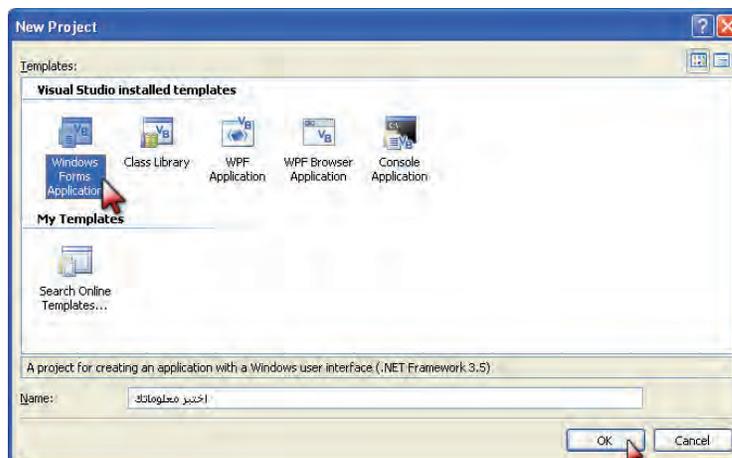
(٦) تطبيق يوضح كيفية التعامل مع المتغيرات

عزيزي المتعلم:

المثال التالي عبارة عن برنامج «بنك المعلومات» ويتكون من نافذة دخول يكتب المستخدم فيها اسمه ونافذتين للأسئلة، وشاشة لعرض النتيجة، ووحدة نمطية Module لتعريف متغيرين على مستوى المشروع «متغير لتخزين الاسم، والآخر لتخزين عدد الأسئلة الصحيحة».

١- أنشئ مشروعاً جديداً وذلك باختيار الأمر New Project من قائمة File لتظهر النافذة التالية،

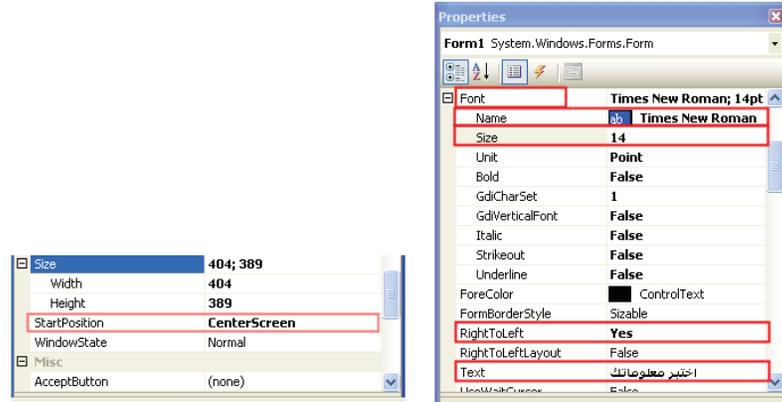
اختر Windows Forms Application



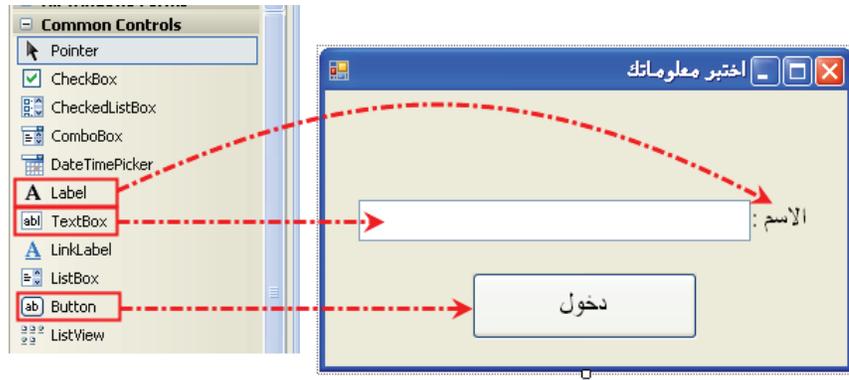
بعد الضغط على الزر «Ok» نلاحظ ظهور النموذج «Form1»

٢- في هذه الخطوة سوف نصمم نافذة الدخول التي بها الاسم كما يلي:

أ - غير خصائص النموذج الأول «Form 1» كما يلي:



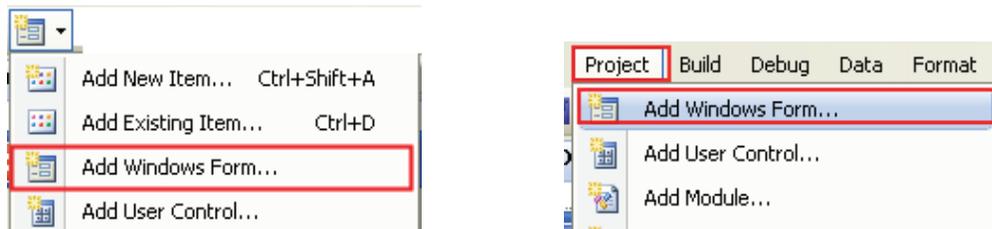
ب- أضف الكائنات التالية للنموذج السابق كما يلي:



وبهذا نكون قد انتهينا من تصميم نافذة الدخول للبرنامج. وسوف نبدأ في تصميم نوافذ الأسئلة ونافذة النتيجة كما يلي:

٣- أضف نموذجاً جديداً للمشروع كما يلي:

أ - اختر الأمر Add Windows Form من القائمة Project أو الأداة



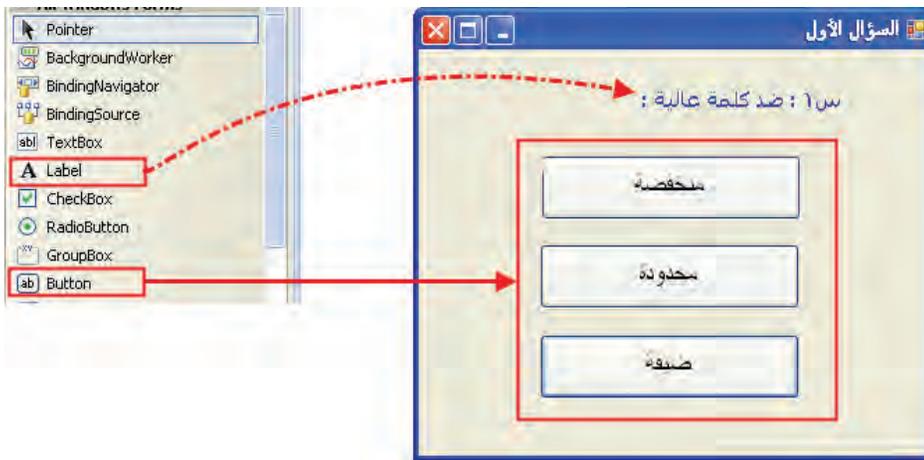
ب- تظهر النافذة التالية .



لاحظ أن اسم النموذج الثاني «Form2». ويظهر اسمه في مستكشف الحل بعد الضغط على Add .

٤- عدّل خصائص النموذج الثاني «Form2» كما في النموذج الأول «Form1» مع تغيير العنوان «Text» إلى السؤال الأول .

٥ - أضف أدوات التحكم المناسبة ليظهر النموذج كما يلي .



٦- كرر الخطوات «٣ ، ٤ ، ٥» لإضافة نموذج ثالث لوضع السؤال الثاني ليظهر كما يلي .

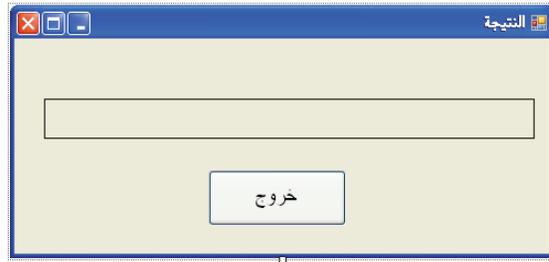


لاحظ أن اسم النموذج الثالث «Form3» ويظهر اسمه في مستكشف الحل .

في النموذج السابق يمكن نسخ ولصق الكائنات بدلاً من إنشائها مرة أخرى.



٧- أضف نموذجاً رابعاً إلى المشروع كما في الخطوة رقم «٣» لإظهار النتيجة، وغير خصائصه وأضف الكائنات المناسبة أداة للعنوان «Label» ليظهر به النتيجة وزر «Button» للخروج كما يلي:



الآن وقد انتهينا من التصميم وقبل البدء في برمجة الكائنات تذكر الآتي:

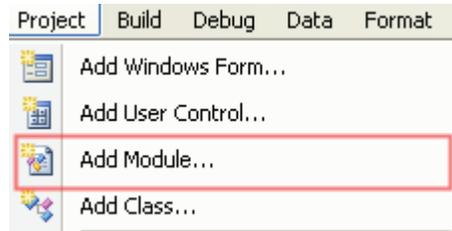
أضفنا إلى المشروع أربعة نماذج كما يلي:

النموذج	الاسم	الوظيفة «الغرض من إنشائه»
الأول	Form1	نافذة دخول للبرنامج
الثاني	Form2	لوضع السؤال الأول
الثالث	Form3	لوضع السؤال الثاني
الرابع	Form4	لإظهار النتيجة .

لبرمجة الكائنات نتبع الآتي:

٨- نحتاج في هذا البرنامج إلى متغيرين، ويكون نطاقهما على مستوى المشروع لنتمكن من رؤيتهما في جميع نماذج المشروع: «الأول لتخزين اسم المتعلم ويكون من نوع String» و «الثاني لتخزين عدد الأسئلة الصحيحة ويكون من نوع Integer»، ولعمل ذلك نضيف Module إلى المشروع كالتالي.

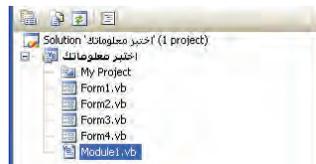
أ- اختر الأمر Add Module من القائمة Project.



تظهر النافذة التالية ومحدد بها النوع كما في الشكل التالي:



نلاحظ ظهور Module1 في مستكشف الحل كما يلي، ويتم فتحه في منطقة التصميم.



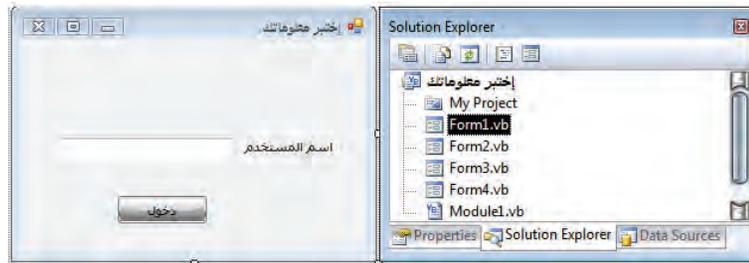
ب- اكتب داخل Module1 التعليمات التالية للإعلان عن متغيرين على مستوى المشروع.

```

Module Module1
    Public user_name As String
    Public grade As Integer
End Module
    
```

لتخزين اسم الطالب
لتخزين درجة الطالب

٩- انتقل إلى النموذج الأول بالضغط المزدوج على اسمه في مستكشف الحل .



١٠- اضغط ضغطاً مزدوجاً على الزر «دخول» لكتابة التعليمات البرمجية اللازمة لتخزين الاسم في المتغير User_name وإخفاء النموذج الحالي وإظهار النموذج الثاني كما يلي:

```

Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        user_name = TextBox1.Text
        Me.Hide()
        Form2.Show()
    End Sub
    
```

لتخصيص اسم الطالب للمتغير User_name
لإخفاء النموذج الحالي
لإظهار النموذج الثاني "Form2"

تستخدم الكلمة المفتاحية Me لتدل على اسم النموذج الحالي .



في التعليمات السابقة استخدمنا الطريقتين

الوظيفة	الطريقة Method
لإخفاء النموذج .	Hide
لإظهار النموذج .	Show

١١- انتقل إلى النموذج الثاني بالضغط المزدوج على اسمه في مستكشف الحل لنبرمج السؤال الأول.

أ- كل سؤال له ثلاثة أزرار واحد فقط يحتوي على الإجابة الصحيحة ، وفي السؤال الأول الإجابة الصحيحة في الزر «Button1» الذي له العنوان «منخفضة» اضغط عليه ضغطاً مزدوجاً واكتب التالي:

```
Button1 Click
Public Class Form2
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Click
        grade = grade + 1
        MsgBox("أحسنت إجابة صحيحة")
        Me.Hide()
        Form3.Show()
    End Sub
End Class
```

ب- اضغط ضغطاً مزدوجاً على الزر الثاني «Button2» الذي له العنوان «محدودة» واكتب التعليمات التالية:

```
Button2 Click
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Click
    MsgBox("احابة خاطنة")
    Me.Hide()
    Form3.Show()
End Sub
```

ج- انسخ التعليمات السابقة وأغلق نافذة التعليمات واضغط ضغطاً مزدوجاً على الزر الثالث «Button3» الذي له العنوان «ضيقة» ثم الصق التعليمات .

```
Button3 Click
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Click
    MsgBox("احابة خاطنة")
    Me.Hide()
    Form3.Show()
End Sub
End Class
```

تستخدم الدالة MsgBox لظهور رسالة للمستخدم .



١٢- انتقل إلى النموذج الثالث وكرر الخطوة رقم «١١ - أ، ب، ج» لبرمجة السؤال الثاني لتظهر التعليمات كما يلي:

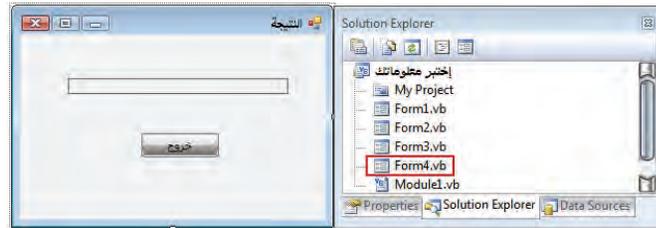
```

Public Class Form3
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        MsgBox("إجابة خاطئة")
        Me.Hide()
        Form4.Show()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        grade = grade + 1
        MsgBox("أجبت إجابة صحيحة")
        Me.Hide()
        Form4.Show()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        MsgBox("إجابة خاطئة")
        Me.Hide()
        Form4.Show()
    End Sub
End Class
    
```

١٣- دعنا الآن نبرمج النموذج الرابع لإظهار النتيجة عند ظهور النموذج .
أ- انتقل إلى النموذج الرابع بالضغط المزدوج على اسمه في مستكشف الحل .



ب- لإظهار النتيجة عند ظهور النموذج اضغط ضغطاً مزدوجاً على النموذج الرابع في أي مكان فارغ واكتب التعليمات التالية:

```

Public Class Form4
    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Label1.Text = " & grade & " على درجة " & user_name & " حصلت يا "
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End Sub
End Class
    
```

يستخدم المعامل «&» في الجمع بين القيم النصية بدلاً من معاملة «+» في القيم الرقمية.



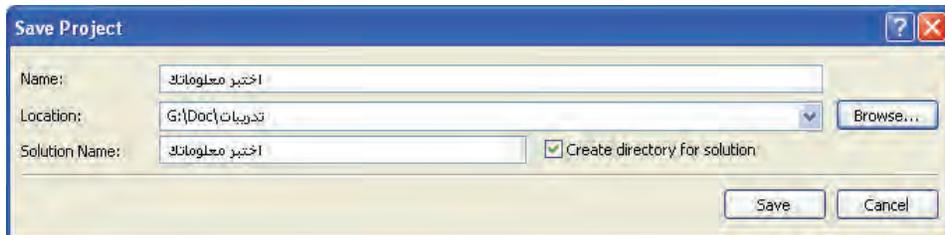
١٤ - لكتابة التعليمات للزر «خروج» اضغط ضغطاً مزدوجاً على الزر واكتب التالي:



```
Private Sub Button13_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button13.Click
    End
End Sub
```

١٥ - جرب البرنامج بالضغط على F5 .

١٦ - احفظ المشروع باسم « اختبار معلوماتك » .



ثانياً: الثوابت Constants

(١) تعريف الثابت

هو مكان داخل الذاكرة تخزن فيه بيانات لا تتغير طيلة فترة تنفيذ البرنامج .

(٢) الإعلان والتخصيص للثابت

يتم الإعلان وتخصيص قيمة للثابت في نافذة التعليمات البرمجية باستخدام تعليمة واحدة Const والتي تخضع للقاعدة التالية:

`Const Name As Type = Value`

حيث :

Name : اسم المتغير .

Type : نوع المتغير .

Value : قيمة الثابت .

أمثلة على تعليمة Const

`Const LastName as string = "محمد أحمد"`

في هذا المثال اسم الثابت هو LastName ونوعه String و قيمته «محمد أحمد».

`Const Pi = 3.14`

في هذا المثال اسم الثابت هو Pi و قيمته 3.14 .

الثوابت لها نفس أنواع المتغيرات (String, Integer, ...).
ونفس النطاقات (ثابت على مستوى المشروع ، ثابت على مستوى النموذج ، ثابت على مستوى الإجراء الفرعي) .



يمكن الاعلان عن الثابت بدون تحديد نوع له وسوف يقوم البرنامج بتحديد النوع المناسب حسب قيمته.



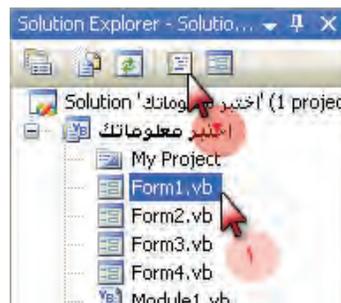
(٣) تطبيق على استخدام الثوابت

في البرنامج السابق «اختبر معلوماتك» نريد أن نضع اسم مصمم البرنامج في نافذة الدخول ، ولعمل ذلك نتبع الخطوات التالية:

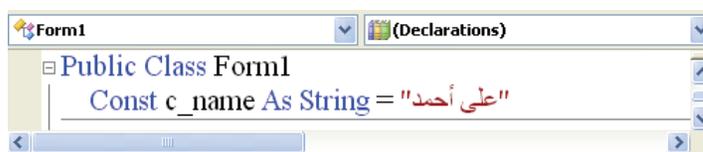
- ١- افتح المشروع السابق «اختبر معلوماتك» باستخدام الأمر «Open Project» من قائمة «File» أو من قائمة المشروعات الأخيرة من نافذة «Start Page».
- ٢- أضف كائنين «Label» للنموذج الأول Form1 كما يلي.



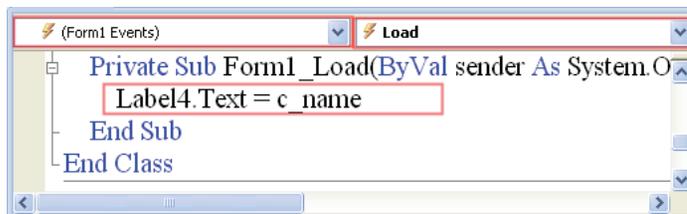
- ٣- اظهر نافذة التعليمات للنموذج «Form1» كما يلي.



- ٤- أعلن عن الثابت «C_name» على مستوى النموذج وأسند له «اسمك» كما يلي:



- ٥- أغلق نافذة التعليمات السابقة ثم اضغط ضغطاً مزدوجاً على النموذج «Form1» في أي مكان فارغ، واكتب التعليمات التالية التي تُظهر قيمة الثابت «اسمك» عند تشغيل البرنامج:



- ٦- احفظ التعديلات على البرنامج وجرب البرنامج.

الفصل الثاني

التفريع

تعلیمة :
إذا...نفذ IF...Then

تعلیمة :
في حالة Select Case



visual Basic



فيما سبق كان البيسك المرئي ينفذ جميع التعليمات البرمجية واحدة تلو الأخرى ، ولكن في كثير من الأحيان نحتاج أن ننفذ مجموعة من التعليمات إذا تحقق شرط معين، وإذا لم يتحقق ننفذ مجموعة تعليمات أخرى ، فمثلاً إذا أردنا من البرنامج أن يظهر رسالة «ناجح» إذا كانت درجة المتعلم أكبر من أو تساوي ٥٠ أو يظهر رسالة «راسب» إذا كانت درجة المتعلم أصغر من ٥٠.

أولاً: تعليمة : إذا... نفذ IF ... Then

تستخدم لتنفيذ تعليمة برمجية إذا تحقق شرط واحد فقط ، ولها عدة صور:

(١) الصورة الأولى تعليمة : إذا... نفذ IF ... Then

If Condition Then Statement

حيث :

Condition شرط منطقي
Statement تعليمة برمجية

الجدول الآتي يوضح عوامل المقارنة:

م	عامل المقارنة	الرمز المعبر عن عامل المقارنة كما يكتب في شبكة تصميم الاستعلام
١	يساوي	=
٢	لا يساوي	< >
٣	أقل من	<
٤	أقل من أو يساوي	< =
٥	أكبر من	>
٦	أكبر من أو يساوي	> =

مثال :

إذا كانت قيمة الاشتراك تساوي ١٠٠ أظهر رسالة ترحيب

IF Subscript = 100 Then MsgBox(“مرحباً”)

(٢) الصورة الثانية تعليمة : إذا... نفذ... وإلا نفذ ...IF ... Then...Else

If condition Then Statement1 Else statement2

حيث :

Condition شرط منطقي
Statement1 التعليمة التي تنفذ في حال تحقق الشرط
Statement2 التعليمة التي تنفذ في حال عدم تحقق الشرط

مثال :

إذا كانت قيمة الاشتراك أكبر من أو تساوي ١٠٠ أظهر مرحباً وإلا أظهر "ممنوع"

IF Subscript >= 100 Then MsgBox("مرحباً") Else MsgBox("ممنوع")

(٣) الصورة الثالثة تعليمة : إذا... نفذ... وإلا نفذ

...IF ... Then...Else لعدة تعليمات :

```
If Condition Then
    Statements Group1
Else
    Statements Group2
End If
```

شرط منطقي
مجموعة التعليمات التي تنفذ في حال تحقق الشرط.
مجموعة التعليمات التي تنفذ في حال عدم تحقق الشرط.

حيث :
Condition
Statements Group1
Statements Group2

مثال :

إذا كانت قيمة الاشتراك أكبر من أو تساوي ١٠٠ أظهر رسالة «مرحباً» ثم «على الرحب والسعة» وإلا أظهر «ممنوع» ثم حاول عندما تحصل على قيمة الاشتراك.

IF Subscript >= 100 Then

MsgBox("مرحباً")

MsgBox("على الرحب والسعة")

Else

MsgBox("ممنوع")

MsgBox("حاول عندما تحصل على قيمة الاشتراك")

End if

لا تصح العبارة السابقة بدون استخدام Else وفي حالة تنفيذ عدة تعليمات لا بد

من End if



تطبيق :

لإضافة كلمة سر للدخول على مشروع «اختبر معلوماتك»، نزيد على النموذج Form1 عنواناً Label5 يحتوي عبارة «كلمة السر»، ومربع نص TextBox2 يستقبل كلمة السر من المستخدم كما في الشكل التالي :



ونضيف الثابت p_word ونخصص له القيمة 11 إلى المنطقة العامة للنموذج الأول كما في الشكل التالي :

```
Public Class Form1
    Const c_name = "علي أحمد"
    Const p_word = 11
```

كما نعدل تعليمات إجراء زر «دخول» مستخدمين تعليمة If كما في الشكل التالي :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal
    user_name = Me.TextBox1.Text
    If TextBox2.Text = p_word Then
        Me.Hide()
        Form2.Show()
    Else
        MsgBox("حاول مرة أخرى وأدخل كلمة السر بشكل صحيح")
    End If
End Sub
```

بعد هذه الإضافة عند تشغيل البرنامج سيختبر محتويات TextBox2 فإذا كانت 11 سيدخلك على الاختبار وإذا كانت لا تساوي 11 فإنه يظهر الرسالة «حاول مرة أخرى وأدخل كلمة السر بشكل صحيح»

ثانياً : تعليمية : في حالة Select case

تستخدم في تنفيذ عدة تعليمات برمجية حسب قيمة متغير ما وتخضع للصورة التالية :

```
Select case V_name
Case Value1
Statements Group1
Case Value2
Statements Group2
...
End Select
```

حيث :

V_name	اسم المتغير .
Value1	القيمة الأولى للمتغير وعند تحققها ينفذ Statements Group1 .
Value2	القيمة الأولى للمتغير وعند تحققها ينفذ Statements Group2 .
...	...

مثال :

لحساب التقديرات نكتب التعليمات التالية لمتغير الدرجة grade

```
Select Case grade
Case 0 to 49.9
MsgBox («راسب»)
Case 50 to 69.9
MsgBox («مقبول»)
Case 70 to 79.9
MsgBox («جيد»)
Case 80 to 89.9
MsgBox («جيد جداً»)
Case 90 to 100
MsgBox («ممتاز»)
End Select
```

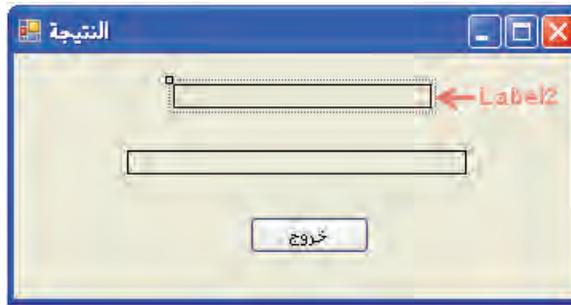
- يمكن كتابة عدة قيم للمتغير في حالة واحدة تفصل بينهما فاصلة عادية Case 20,40,60 .
- يمكن اختبار قيمة للمتغير تقع بين قيمتين Case 8 to 10 ، الشرط صحيح للقيم ٨ أو ٩ أو ١٠ .
- يمكن استخدام احد معاملات المقارنة مثل (<، >) بالصورة التالية Case is < 100
تعني أن قيمة متغير Select أقل من ١٠٠ .



تطبيق :

إضافة عبارة تعزيز في نموذج النتيجة لمشروع «اختبر معلوماتك» من خلال :

١ - إضافة عنوان Label2 لنموذج النتيجة Form4 .



٢ - إضافة تعليمة Select case التي تختبر قيمة متغير الدرجة grade وتضع عبارة التعزيز

المناسبة في العنوان Label2 لإجراء تحميل Form4

```

Form4.vb* Form4.vb [Design]* Start Page
(Form4 Events) Load
Public Class Form4
    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Select Case grade
            Case 0
                Label2.Text = "خط أوفر المرة القادمة"
            Case 1
                Label2.Text = "أحسن"
            Case 2
                Label2.Text = "رائع"
        End Select
        Label1.Text = " على درجة " & grade & " حصلت يا " & user_name
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        End
    End Sub
End Class
    
```



المشروع



visual Basic



عزيزي المتعلم

لقد اكتسبت في الجزء السابق من هذا الكتاب مجموعة من المعارف والمهارات عن لغة البرمجة المرئية Visual Basic 2008، وطبقت بعض التدريبات التي أعدت بهدف تعزيز تلك المهارات، وقد تم التخطيط لتقسيم هذه التدريبات ليتفق كل قسم مع ما درسته في كل حصة دراسية على حدة، والآن عليك أن تخطط لإنتاج مشروع متكاملٍ توظف فيه كل قدراتك على استخدام لغة البرمجة المرئية Visual Basic 2008، وتعزز وتعمق من خلاله تلك المعارف والمهارات التي تعلمتها .

أولاً : أهداف المشروع

إن الهدف الأساسي من المشروع هو استخدامه كأداة تمكنك من الاستفادة من المهارات التي درستها وتنمي من خلاله العديد من المهارات منها :

- اكتساب مهارات العمل الجماعي التعاوني .
- القدرة على الاتصال بالآخرين وتجميع المعلومات اللازمة لإنتاج مشروعك .
- القدرة على الربط بين المعلومات ، وتنظيمها وترتيبها للاستفادة منها .
- القدرة على توظيف المهارات التي تعلمتها للاستفادة منها .
- القدرة على توظيف مهاراتك في خدمة المجتمع من حولك .
- القدرة على التحليل و حل المشكلات .
- القدرة على الابتكار .

ثانياً : مجال المشروع

يهدف المشروع في الصف الثاني عشر إلى تحقيق الفائدة من المهارات التي درستها لخدمة المجتمع من حولك بإنتاج برامج تخدم إحدى الموضوعات التالية :

- برنامج مسابقات «اختبر معلوماتك» في أحد المجالات العلمية أو الثقافية .
- برنامج موسوعة علمية تخدم أحد المجالات العلمية أو الثقافية .
- برنامج يخدم المدرسة (كعارض لأنشطة المتعلمين أو المعلمين ... إلخ) .
- أحد المشاريع الأخرى التي يتم الاتفاق عليها بين مجموعة العمل ومعلم الفصل .

ثالثاً : مراحل إعداد المشروع

بعد تقسيم المجموعات واختيارك للفريق المناسب للعمل ، وتحديد موضوع المشروع الذي سيتم إنتاجه، يجب عليك تقسيم العمل إلى مراحل لإنتاجه ، لضمان التنسيق وتقسيم العمل بين أفراد الفريق ، ويمكننا إيجاز هذه المراحل في التالي :

المرحلة الأولى (الحصول على المعلومات)

تعتبر هذه المرحلة من أهم مراحل إعداد البرامج ، حيث إن الدقة في الحصول على المعلومات عن المشروع ستساعدنا في التخطيط الجيد لإنتاجه، ويمكن أن نرتب عملية الحصول على المعلومة فيما يلي :

- ١ - اتفاق الفريق مع المسؤول على الجهة التي سيعد من أجلها البرنامج .
- ٢ - تحديد الهدف من إنشاء البرنامج من خلال مناقشة المسؤول .

المرحلة الثانية (تحليل المشكلات)

- ١ - حصر البيانات الواجب إدخالها للبرنامج (مدخلات البرنامج) .
- ٢ - تحديد المعلومات المطلوبة من قاعدة البرنامج (مخرجات البرنامج) .
- ٣ - تحديد عمليات المعالجة ، والتعليمات البرمجية التي يلزم إجراؤها على المدخلات .

المرحلة الثالثة (التخطيط لإنشاء البرنامج من خلال البيسك المرئي)

- ١ - تحديد الشكل العام للبرنامج ونافذته الرئيسية .
- ٢ - تحديد عدد النوافذ الفرعية للبرنامج ، وشكلها .
- ٣ - تحديد عدد الكائنات المطلوب إدراجها لنوافذ البرنامج .
- ٤ - تحديد الخصائص المطلوبة لكل كائن من كائنات البرنامج .
- ٥ - تحديد الأحداث المطلوبة لتفاعل المستخدم مع البرنامج .
- ٦ - إعداد دليل العمل الذي ستحفظ عليه مشروعك .
- ٧ - كتابة التعليمات البرمجية اللازمة للكائنات .

المرحلة الرابعة (إعداد البرنامج)

- ١ - إنشاء النماذج المطلوبة .
- ٢ - إضافة الكائنات للنماذج .
- ٣ - تعديل خصائص الكائنات لتناسب الوظيفة المطلوبة منها .

المرحلة الخامسة (اختبار البرنامج)

- ١ - اختبار البرنامج للتأكد من صحة التعليمات البرمجية .
- ٢ - تصحيح الأخطاء البرمجية إن وجدت .

المرحلة السادسة (تحويل المشروع لملف تنفيذي)

- ١ - تحديد الدليل المراد حفظ الملف التنفيذي به .
- ٢ - تحويل المشروع لملف تنفيذي .

المرحلة السابعة (عرض ومناقشة المشروع)

- ١ - تحديد أحد أفراد المجموعة ممن يتمتع بمهارة التحدث ، لعرض المشروع على بقية المجموعات .
- ٢ - استطلاع آراء بقية المجموعات عن المشروع ومدى تحقيقه للهدف المعد من أجله .



كراس المتعلم



visual Basic

الباب الأول مدخل إلى البيسك المرئي

الأول	الدرس
.....	اليوم
..... / /	التاريخ
مفاهيم أساسية	عنوان الدرس
<ul style="list-style-type: none">● تعريف البرنامج .● تعريف لغة البرمجة .● أهمية لغات البرمجة .● أقسام لغات البرمجة .	بنود الدرس
ورقة عمل (١)	التطبيق
.....	التقويم الصفّي
.....	التقويم اللاصفي

ورقة عمل (١)

من خلال مجلد «تدريبات» :



١ - شغل الملف «بيسك»

٢ - اختر الرابط (مفاهيم أساسية).

- تتبع الروابط في البرنامج، ونفذ ما تطلبه منك.



الدرس	الثاني
اليوم
التاريخ / /
عنوان الدرس	واجهة البيسك المرئي ٢٠٠٨
بنود الدرس	<ul style="list-style-type: none"> ● تعريف البيسك المرئي ٢٠٠٨. ● مزايا البيسك المرئي ٢٠٠٨ . ● تشغيل البيسك المرئي ٢٠٠٨. ● فتح مشروع سبق إنشاؤه . ● عناصر واجهة البيسك المرئي ٢٠٠٨. ● التحكم في نوافذ الواجهة .
التطبيق	ورقة عمل (٢)
التقويم الصفي
التقويم اللاصفي

ورقة عمل (٢)

من خلال مجلد «تدريبات» :



١ - شغل الملف «بيسك»

٢ - اختر الرابط (واجهه البيسك المرئي).

- تتبع الروابط في البرنامج ، ونفذ ما تطلبه منك .



الباب الثاني : مراحل بناء برنامج

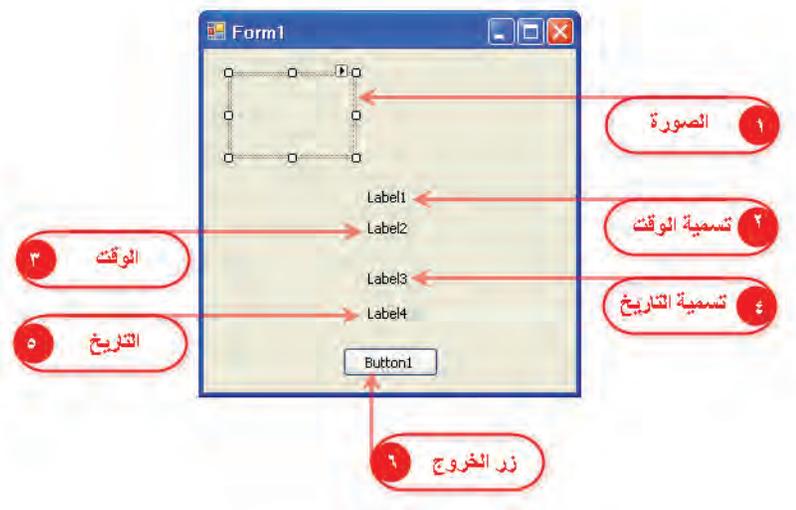
الدرس	الثالث
اليوم
التاريخ / /
عنوان الدرس	بداية بناء برنامج
بنود الدرس	<ul style="list-style-type: none">● التخطيط للبرنامج.● إنشاء مشروع جديد.● إضافة أدوات تحكم.
التطبيق	ورقة عمل (٣)
التقويم الصفّي
التقويم اللاصقي

ورقة عمل (٣)

- ١ - شغل البيسك المرئي ٢٠٠٨ ،
- ٢ - أنشئ حلاً جديداً باسم "Welcome" .
- ٣ - احفظ المشروع باسم "Welcome" .
- ٤ - أضف للنموذج كائناته بالأعداد والأنواع المبينة في الجدول الآتي :

عدد الكائنات	الكائن	وصف الكائن
١	Button	زر أمر
١	PictureBox	صندوق صورة
٤	Label	عنوان

بحيث يصبح النموذج بالصورة التالية :



- ٥ - احفظ التعديلات ثم اغلق البيسك المرئي ٢٠٠٨ .

الباب الثاني : مراحل بناء برنامج

الدرس	الرابع
اليوم
التاريخ / /
عنوان الدرس	ضبط عناصر البرنامج
بنود الدرس	<ul style="list-style-type: none">● تعريف الخاصية.● تغيير خاصية كائن أثناء التصميم.
التطبيق	ورقة عمل (٤)
التقويم الصفّي
التقويم اللاصفي

ورقة عمل (٤)

- ١ - شغل البيسك المرئي ٢٠٠٨ .
- ٢ - افتح المشروع السابق «Welcome» .
- ٣ - اضبط خصائص كائنات النموذج بحيث يصبح كما في الصورة التالية:



مع العلم بأن :

(أ) الصورة الموجودة في مجلد «التدريبات» باسم «Clock» .

(ب) الخط Tahoma وحجمه ١٠ ونمطه Bold .

٤ - احفظ التعديلات ثم أغلق البيسك المرئي ٢٠٠٨ .

الدرس	الخامس
اليوم
التاريخ / /
عنوان الدرس	كتابة التعليمات البرمجية
بنود الدرس	<ul style="list-style-type: none"> ● تعريف التعليمات البرمجية. ● تعريف الحدث Event. ● تعريف محرر التعليمات البرمجية. ● اختبار البرنامج. ● بناء البرنامج.
التطبيق	ورقة عمل (٥)
التقويم الصفّي
التقويم اللاصقي

ورقة عمل (٥)

- ١ - شغل البيسك المرئي ٢٠٠٨ ،
- ٢ - افتح المشروع السابق «Welcome» .
- ٣ - اكتب التعليمات البرمجية التي تجعل زر «خروج» ينهي البرنامج .
- ٤ - اكتب التعليمات البرمجية لتغيير العنوان Label2 ليعرض الوقت الحالي عند بدء التشغيل .
label2.text = Timestring
- ٥ - اكتب التعليمات البرمجية لتغيير العنوان Label4 ليعرض التاريخ الحالي عند التشغيل .
label4.text = Datestring
- ٦ - اختبر البرنامج .
Debug
- ٧ - ابن الملف التنفيذي للبرنامج على القرص الخاص بك .
Build > Build Welcome
- ٨ - احفظ التعديلات ثم أغلق البيسك المرئي ٢٠٠٨ .

السادس	الدرس
.....	اليوم
..... / /	التاريخ
المتغيرات Variable	عنوان الدرس
<ul style="list-style-type: none"> ● تعريف المتغير . ● أنواع المتغيرات . ● الإعلان عن المتغيرات ● تخصيص قيم للمتغيرات . 	بنود الدرس
ورقة عمل (٦)	التطبيق
.....	التقويم الصفّي
.....	التقويم اللاصفي

ورقة عمل (٦)

- ١ - شغل البيسك المرثي ٢٠٠٨ .
- ٢ - أنشئ مشروعاً جديداً باسم "بنك المعلومات" .
- ٣ - احفظ مشروعك بنفس الاسم على المجلد الخاص بك .
- ٤ - عدل خصائص النموذج «Form1» كما يلي :

الخاصية	القيمة
Name	Form1
Text	بنك المعلومات
Font-Size	14
RightToLeft	Yes
StartPosition	CenterScreen

- ٥ - أضف للنموذج الأول الكائنات التالية ليظهر كما يلي :



- ٦ - احفظ التعديلات التي تمت على المشروع ، ثم أغلق البيسك المرثي ٢٠٠٨ .

السابع	الدرس
.....	اليوم
..... / /	التاريخ
تابع المتغيرات Variable	عنوان الدرس
<ul style="list-style-type: none"> ● نطاق المتغيرات . ● تطبيق يوضح كيفية التعامل مع المتغيرات : (أ) بناء النماذج . (ب) إضافة الكائنات لنماذج التطبيق . 	بنود الدرس
ورقة عمل (٧)	التطبيق
.....	التقويم الصفي
.....	التقويم اللاصفي

ورقة عمل (٧)

- ١ - شغل البيسك المرئي ٢٠٠٨ .
- ٢ - افتح مشروع «بنك المعلومات» .
- ٣ - أضف نموذجاً ثانياً للمشروع .
- ٤ - اضبط خصائص النموذج كما يلي :

الكائن	الخاصية	القيمة
Form	Name	Form2
	Text	السؤال الأول
	RightToLeft	Yes
	Font-size	١٤

- ٥ - أضف الكائنات التالية إلى النموذج ليظهر كما يلي :

الكائن	الخاصية	القيمة
label1	Name	Label1
	Text	س ١ : لتخزين عنوان المتعلم نستخدم متغيراً من نوع
Button1	Name	Button1
	Text	Date
Button2	Name	Button2
	Text	String
Button3	Name	Button3
	Text	Integer

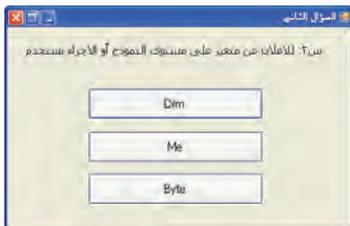
- ٦ - أضف نموذجاً ثالثاً للمشروع واضبط خصائصه .

كما في النموذج السابق . أضف الكائنات المناسبة للنموذج ليظهر كما يلي . لاحظ خاصية Text للنموذج

- ٧ - أضف نموذجاً رابعاً للمشروع واضبط خصائصه .

كما في النماذج السابقة . ليظهر كما يلي :

- ٨ - احفظ التعديلات التي تمت على المشروع ، ثم اغلق البيسك المرئي ٢٠٠٨ .



الدرس	الثامن
اليوم
التاريخ / /
عنوان الدرس	تابع المتغيرات Variable
بنود الدرس	<ul style="list-style-type: none"> ● تطبيق يوضح كيفية التعامل مع المتغيرات : <p>(ج) كتابة التعليمات البرمجية لنماذج التطبيق .</p>
التطبيق	ورقة عمل (٨)
التقويم الصفّي
التقويم اللاصقي

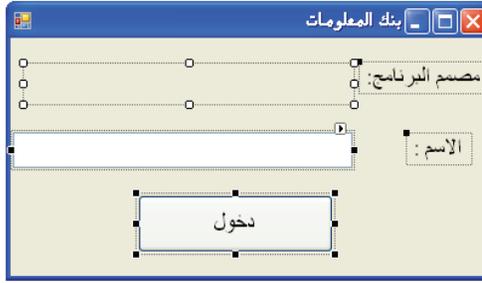
ورقة عمل (٨)

- ١ - شغل البيسك المرئي ٢٠٠٨ .
- ٢ - افتح مشروع «بنك المعلومات» .
- ٣- أضف وحدة نمطية «Module» للمشروع وأعلن عن متغيرين أحدهما S_name من نوع String والثاني S_grade من نوع Integer.
- ٤- انتقل إلى النموذج الأول «Form1» واكتب التعليمات البرمجية للزر «دخول» لتخصيص اسم المتعلم للمتغير «S_name» وإخفاء النموذج الأول وإظهار النموذج الثاني .
- ٥- انتقل إلى النموذج الثاني «Form2» واكتب التعليمات البرمجية للزر الثاني «String» بحيث يضاف واحد للمتغير S_grade ويتم إخفاء النموذج الحالي وإظهار النموذج الثالث .
- ٦- اكتب التعليمات البرمجية للزرين الأول والثالث لإخفاء النموذج الحالي وإظهار النموذج الثالث مع إظهار رسالة تفيد بأن الإجابة غير صحيحة.
- ٧- انتقل للنموذج الثالث «Form3» واكتب التعليمات البرمجية للأزرار الثلاثة، كما في النموذج الثاني مع مراعاة الإجابة الصحيحة.
- ٨- انتقل للنموذج الرابع «Form4» واكتب التعليمات البرمجية على النموذج لتظهر النتيجة عند فتح النموذج.
- ٩- انتقل للنموذج الرابع «Form4» واكتب التعليمات البرمجية على الزر «خروج» لإنهاء البرنامج .
- ١٠- اختبر البرنامج.
- ١١- احفظ التعديلات التي تمت على المشروع ثم أغلق البيسك المرئي ٢٠٠٨ .

التاسع	الدرس
.....	اليوم
..... / /	التاريخ
الثوابت Constants	عنوان الدرس
<ul style="list-style-type: none"> ● تعريف الثابت . ● الإعلان و التخصيص للثابت . ● تطبيق على استخدام الثوابت . 	بنود الدرس
ورقة عمل (٩)	التطبيق
.....	التقويم الصفي
.....	التقويم اللاصفي

ورقة عمل (٩)

- ١- شغل البيسك المرئي ٢٠٠٨ .
- ٢- افتح مشروع «بنك المعلومات».
- ٣- انتقل إلى النموذج الأول «Form1» وأضف الكائنات المناسبة ليظهر النموذج كما يلي :



- ٤- أضف ثابتاً على مستوى النموذج باسم «S_design» وخصص له القيمة « اسمك » .
- ٥- اجعل اسمك يظهر في بداية تشغيل النموذج .
- ٦- اختبر البرنامج .
- ٧- احفظ التعديلات التي تمت على المشروع ثم أغلق البيسك المرئي ٢٠٠٨ .

العاشر	الدرس
.....	اليوم
..... / /	التاريخ
تعلیمة : إذا... نفذ IF ... then	عنوان الدرس
<ul style="list-style-type: none"> ● الصورة الأولى لتعلیمة : إذا... نفذ IF ... Then ● الصورة الثانية لتعلیمة إذا... نفذ... وإلا نفذ IF ... Then...Else ● الصورة الثالثة لتعلیمة إذا... نفذ... وإلا نفذ ...IF ... Then...Else <p>لعدة تعلیمات ،</p> <ul style="list-style-type: none"> ● تطبيق على تعلیمة إذا... نفذ IF ... Then 	بنود الدرس
ورقة عمل (١٠)	التطبيق
.....	التقويم الصفی
.....	التقويم اللاصفی

ورقة عمل (١٠)

- ١ - شغل البيسك المرئي ٢٠٠٨ ،
- ٢ - افتح مشروع «بنك المعلومات» .
- ٣- أضف للنموذج الأول عنوان « أدخل كلمة السر» .
- ٤ - أضف للنموذج الأول صندوق نص يستقبل كلمة السر .
- ٥ - أضف التعليمات البرمجية لاختبار كلمة السر عند الضغط على زر دخول ولتكن قيمتها ٢٠٠ .
- ٦ - اختبر البرنامج للتأكد من فاعلية كلمة السر .
- ٧ - احفظ التعديلات التي تمت على المشروع ثم أغلق البيسك المرئي ٢٠٠٨ .

الحادي عشر	الدرس
.....	اليوم
..... / /	التاريخ
Select case : تعليمة في حالة	عنوان الدرس
<ul style="list-style-type: none"> ● الصورة العامة لتعليمة في حالة Select case. ● تطبيق على تعليمة في حالة Select case. 	بنود الدرس
ورقة عمل (١١)	التطبيق
.....	التقويم الصفّي
.....	التقويم اللاصفي

ورقة عمل (١١)

- ١ - شغل البيسك المرئي ٢٠٠٨،
- ٢ - افتح مشروع «بنك المعلومات» .
- ٣ - أضف عنوان Label2 للنموذج النتيجة Form4.
- ٤ - مستعين بتعليمة Select case أضف إلى التعليمات البرمجية لحدث Form Load لكي يظهر داخل العنوان Label2 :
 - أ - عبارة التعزيز «رائع» إذا حصل المتعلم على درجتين .
 - ب - عبارة التعزيز «أحسن» إذا حصل المتعلم على درجة واحدة .
 - ج - عبارة التعزيز «حظ أوفر المرة القادمة» إذا لم يحصل المتعلم على أي درجة .
- ٥ - اختبر المشروع على ثلاث مرات :
 - أ - أجب عن جميع الأسئلة بشكل صحيح ، ماذا تجد عند الوصول للنموذج الأخير؟
.....
 - ب - أجب عن سؤال واحد بشكل صحيح وعن الآخر بشكل غير صحيح، ماذا تجد عند الوصول للنموذج الأخير؟
.....
 - ج - أجب عن جميع الأسئلة بشكل غير صحيح ، ماذا تجد عند الوصول للنموذج الأخير؟
.....
- ٦ - احفظ التعديلات ثم أغلق البيسك المرئي ٢٠٠٨ .

الباب الثالث : التعليمات البرمجية الأساسية

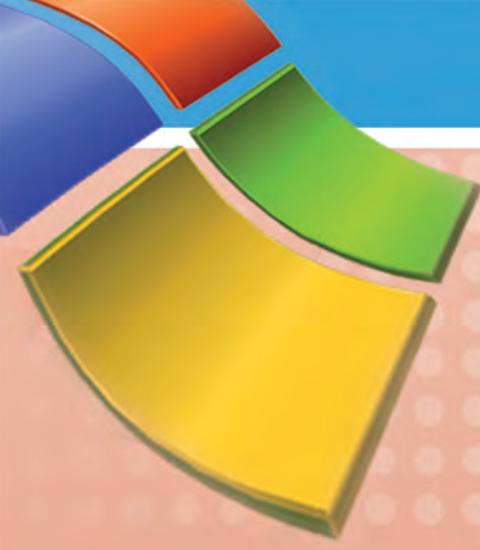
الدرس	الثاني عشر
اليوم
التاريخ / /
عنوان الدرس	المشروع
بنود الدرس	<ul style="list-style-type: none"> ● اختيار أحد المشاريع من وحدة المشروع بالتعاون مع المعلم. ● تخطيط البرنامج. ● تصميم البرنامج على البيسك المرئي ٢٠٠٨. ● تشغيل واختيار البرنامج.
التطبيق	مناقشة البرنامج
التقويم الصفي
التقويم اللاصفي

التقويم



visual Basic



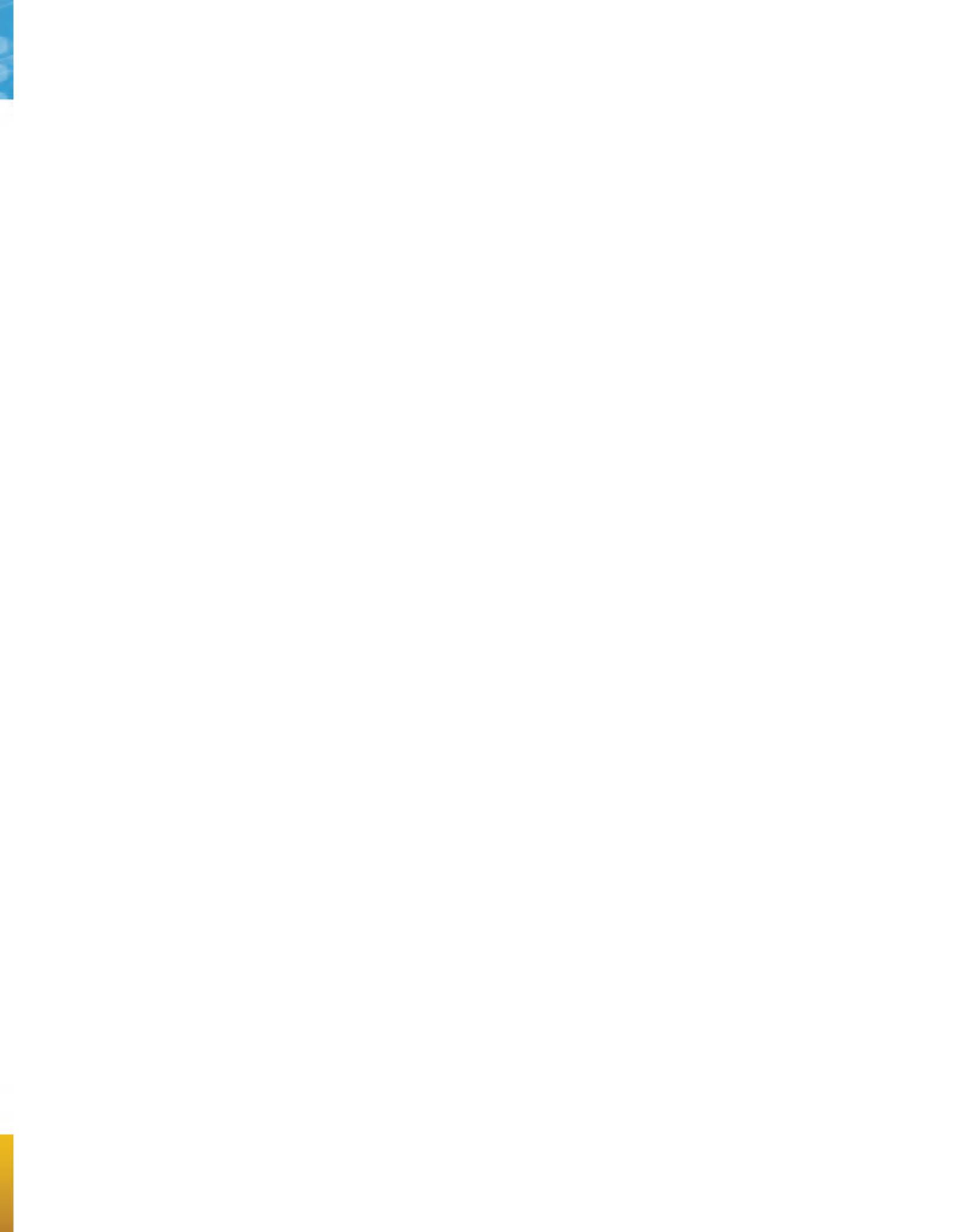


الباب الأول

مدخل إلى البيسك المرئي



visual Basic



الأسئلة الموضوعية

أولاً: في البنود المرقمة من (٧-١) توجد عبارات صحيحة وأخرى غير صحيحة، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة، (ب) إذا كانت العبارة غير صحيحة:

١	البرنامج هو مجموعة من التعليمات المنطقية يصدرها الإنسان للحاسوب للقيام بمهمة محددة.	أ	ب
٢	المبرمج هو الشخص الذي يقوم بتشغيل برامج الحاسوب .	أ	ب
٣	لغة البرمجة هي أداة تساعدنا على بناء البرامج بجميع أنواعها .	أ	ب
٤	تعتمد لغة الآلة في كتابة تعليماتها على لغة قريبة من لغة الإنسان .	أ	ب
٥	تعد لغة التجميع Assembly Language من لغات المستوى الأدنى .	أ	ب
٦	لغات البرمجة النصية Script languages تعتمد على عبارات إنجليزية اصطلاحية للدلالة على الأوامر التي توجه الحاسوب .	أ	ب
٧	لغات البرمجة المرئية تبني برامج ذات واجهة رسومية ولا تحتاج لكتابة أوامر نصية .	أ	ب

ثانياً: في البنود المرقمة من (٣-١) توجد قائمتان (الأولى والثانية)، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

الإجابة	القائمة الأولى من لغات البرمجة	القائمة الثانية تسمى
١	اللغة التي اعتمدت على أرقام النظام الثنائي (١،٠)	اللغة الإنجليزية .
٢	اللغة التي اعتمدت على اختصارات مثل (Int ، Add) في كتابة تعليماتها .	لغة الآلة .
٣	اللغة التي تعتمد على عبارات إنجليزية اصطلاحية للدلالة على الأوامر التي توجه الحاسوب .	لغات المستوى العاليي .
		لغة التجميع .

الفصل الأول : مفاهيم أساسية

ثالثاً: في البنود المرقمة من (١ - ٥) لكل بند أربعة اختيارات ، واحد فقط منها صحيح ، اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها :

١ مجموعة التعليمات المنطقية التي يوجه المبرمج بها الحاسوب للقيام بمهمة محددة تسمى :			
أ	لغة البرمجة .	ب	لغة بيسك .
ج	البرنامج .	د	لغة الآلة .
٢ أداة تساعدنا على بناء البرامج بجميع أنواعها تسمى .			
أ	لغة البرمجة .	ب	الكيان البرمجي .
ج	البرنامج .	د	الكيان المادي .
٣ لغة البرمجة التي اعتمدت على أرقام النظام الثنائي (١،٠) في كتابة التعليمات البرمجية هي :			
أ	لغة بيسك .	ب	لغة التجميع .
ج	لغة الآلة .	د	لغات المستوى العالي .
٤ لغات البرمجة التي تعتمد فقط على عبارات انجليزية اصطلاحية للدلالة على الأوامر التي توجه الحاسوب هي :			
أ	لغة البرمجة النصية .	ب	برامج النظم .
ج	لغة البرمجة المرئية .	د	البرامج الخدمية .
٥ من لغات المستوى العالي :			
أ	لغة الآلة .	ب	نظام التشغيل .
ج	لغة التجميع :	د	لغة بيسك .

الفصل الأول : مفاهيم أساسية

رابعاً: في البنود المرقمة من (١-٤) عبارات تحتوي كل منها على فراغ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة :

١	مجموعة من التعليمات المنطقية التي يوجه المبرمج بها الحاسوب للقيام بمهمة محددة
٢	أداة تساعدنا على بناء البرامج بجميع أنواعها تسمى
٣	لغة البرمجة التي اعتمدت في كتابة تعليماتها على لغة قريبة من لغة الإنسان تسمى
٤	لغة البرمجة التي تبني برامج ذات واجهة رسومية وتخفي خلفها أوامر نصية توجه الحاسوب تسمى

الأسئلة المقالية

أجب عن الأسئلة الآتية:

١- عرّف كلاً من :

(١) البرنامج :

(٢) لغة البرمجة :

٢- قسّم لغات البرمجة إلى مستوياتها ، مع كتابة نبذة عن كل منها :

(١)

(٢)

٣- عدّد اثنتين من لغات البرمجة ذات المستوى الأدنى :

(١)

(٢)

٤- وضح المقصود بكل من :

(١) لغات البرمجة النصية Script languages

(٢) لغات البرمجة المرئية

الأسئلة الموضوعية

أولاً: في البنود المرقمة من (١ - ١٠) توجد عبارات صحيحة وأخرى غير صحيحة، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة، (ب) إذا كانت العبارة غير صحيحة:

١	من مزايا البيسك المرئي سهولة اكتشاف الأخطاء البرمجية .	أ	ب
٢	يمكن من خلال البيسك المرئي إعداد تطبيقات النوافذ فقط .	أ	ب
٣	تظهر أسماء المشاريع الأخيرة التي تم التعامل معها أسفل نافذة Getting Started .	أ	ب
٤	لفتح مشروع سبق إنشاؤه يستخدم الأمر Open File من قائمة File .	أ	ب
٥	صندوق الأدوات Toolbox عبارة عن نافذة يمكن من خلالها إضافة الكائنات المختلفة للنموذج .	أ	ب
٦	تظهر جميع كائنات صندوق الأدوات Toolbox داخل مجموعة واحدة .	أ	ب
٧	لإظهار صندوق الأدوات Toolbox بعد إغلاقه يجب إغلاق البيسك المرئي وفتحه مرة أخرى .	أ	ب
٨	تتشارك كائنات البيسك المرئي في جميع الخصائص .	أ	ب
٩	نافذة Solution Explorer تظهر بها جميع محتويات المشروع من النماذج Forms والوحدات النمطية . Module	أ	ب
١٠	يمكن نقل نافذة الخصائص Properties إلى أي مكان وهي في وضع الإظهار الدائم .	أ	ب

الفصل الثاني : واجهة البيسك المرئي ٢٠٠٨

ثانياً: في البنود المرقمة من (١-٦) توجد قائمتان (الأولى والثانية) ، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

من خلال دراستك لصفحة البدء عند تشغيل البيسك المرئي Visual Basic 2008 .

القائمة الثانية يستخدم الجزء		القائمة الأولى الوظيفة		الإجابة
Getting Started .	أ	تعرض آخر مشاريع تم التعامل معها لفتحها بصورة سريعة دون الحاجة لمعرفة موقعها .	١	
Recent قائمة المشاريع الأخيرة .projects	ب	تظهر فيها أخبار لغة البيسك المرئي وتحديث تلقائياً أثناء الاتصال بالانترنت .	٢	
الأخبار .News	ج	تعطي معلومات مبسطة عن استخدام البيسك المرئي للمستخدم المبتدئ .	٣	
إعلانات Headlines	د			

من خلال دراستك لعناصر واجهة البيسك المرئي Visual Basic 2008 .

القائمة الثانية يستخدم الجزء		القائمة الأولى الوظيفة		الإجابة
Solution Explorer نافذة	أ	النافذة التي يمكن من خلالها إضافة الكائنات المختلفة للنموذج ،	٤	
ToolBox	ب	النافذة التي يمكن من خلالها ضبط أوصاف الكائنات،	٥	
Headlines	ج	النافذة التي تظهر بها جميع محتويات المشروع من النماذج Forms والوحدات النمطية . Modules	٦	
Properties	د			

الفصل الثاني : واجهة البيسك المرئي ٢٠٠٨

ثالثاً: في البنود المرقمة من (١ - ٧) لكل بند أربعة اختيارات، واحد فقط منها صحيح، اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها:

١ أي مما يلي ليس من مزايا لغة البيسك المرئي Visual basic 2008 :			
أ	لغة تستجيب للأحداث .	ب	سهولة اكتشاف الأخطاء البرمجية .
ج	لغة تعتمد على كائنات رسومية .	د	تتقيد بالعديد من القواعد مثل طبيعة الحروف كبيرة أو صغيرة .
٢ يمكن من خلال لغة البيسك المرئي Visual basic 2008 إنتاج تطبيقات لـ :			
أ	النوافذ .	ب	الويب .
ج	الهاتف النقال .	د	جميع ما سبق صحيح .
٣ في صفحة البدء Start Page الجزء الذي يعرض آخر مشاريع تم التعامل معها لفتحها بصورة سريعة يسمى:			
أ	Getting Started	ب	News
ج	Recent projects	د	Headlines
٤ يمكن فتح مشروع سبق إنشاؤه من خلال :			
أ	الأمر Open Project من قائمة «File»	ب	الضغط على اسمه في قائمة Recent projects
ج	من الجزء Open Project في صفحة البدء .	د	جميع ما سبق صحيح .
٥ عند فتح مشروع ، يظهر في صفحة تسمى :			
أ	Project	ب	Start Page
ج	Design	د	Form

الفصل الثاني : واجهة البيسك المرئي ٢٠٠٨

٦ من عناصر واجهة البيسك المرئي النافذة التي تعرض مكونات المشروع من النماذج Forms ووحدات Modules تسمى :			
Properties	ب	Solution Explorer	أ
Toolbox	د	Start Page	ج
٧ لتثبيت ظهور النوافذ بدلاً من الإخفاء التلقائي يستخدم :			
	ب		أ
	د		ج

رابعاً: في البنود المرقمة من (١ - ٧) عبارات، تحتوي كل منها على فراغ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة :

١ عند تشغيل بيسك المرئي Visual Basic 2008 تظهر لنا النافذة الرئيسية محتوية على	
٢ تظهر داخل الجزء Recent Projects ويمكن فتحها بصورة سريعة .	
٣ لفتح مشروع سبق إنشاؤه يستخدم من قائمة «File» الأمر	
٤ يتم فتح المشروع داخل صفحة تسمى	
٥ يمكن إضافة الكائنات المختلفة إلى النموذج من خلال	
٦ يمكن إظهار صندوق الأدوات بعد إغلاقه من قائمة View باستخدام الأمر	
٧ يمكن إظهار نافذة مستكشف الحل بعد إغلاقها من قائمة View باستخدام الأمر	

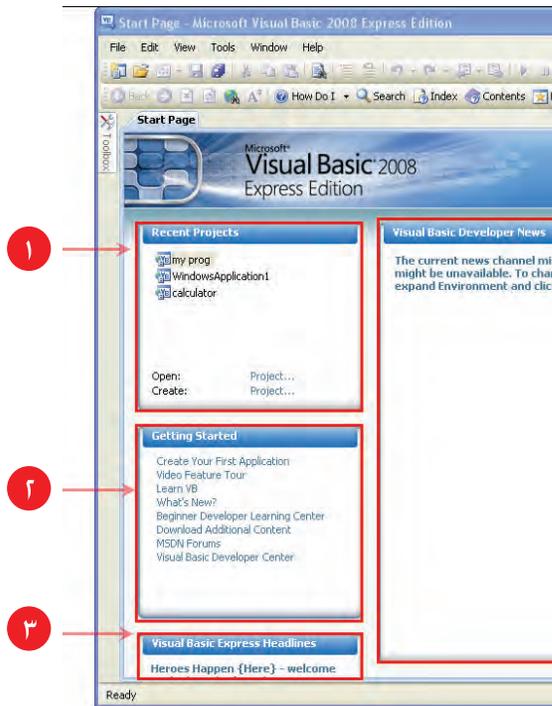
الأسئلة المقالية

أجب عن الأسئلة الآتية:

١- عدد ثلاثاً من مزايا البيسك المرئي Visual basic 2008:

- (١)
- (٢)
- (٢)

٢- من خلال دراستك لصفحة البدء Start Page في نافذة البيسك المرئي Visual Basic 2008، ادرس الشكل الآتي ثم أكمل البيانات الناقصة على الشكل :



الوظيفة	اسم الجزء	الرقم
.....		١
.....		٢
.....		٣

٣- من خلال دراستك لفتح مشروع سبق إنشاؤه ، عدد طرق فتح المشروع :

..... (١)

..... (٢)

..... (٢)

٤- اذكر وظيفة النوافذ التالية :

(١) صندوق الأدوات Toolbox

.....

(٢) نافذة Properties

.....

(٣) نافذة Solution Explorer

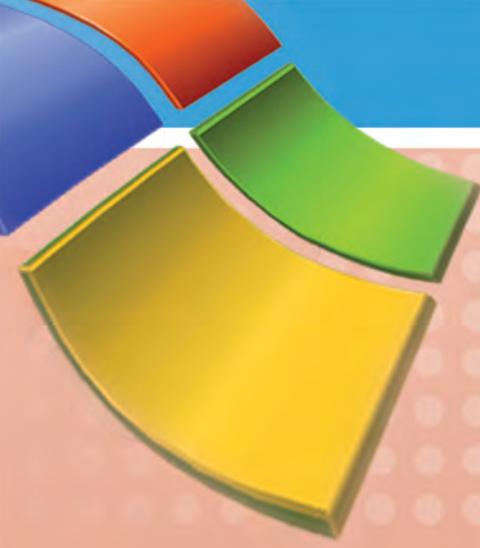
.....

٥- من خلال دراستك للتحكم في نوافذ واجهة البيسك المرئي أكمل الجدول الآتي بما يناسبه :

الأداة	الوظيفة





الباب الثاني

مراحل بناء البرنامج



visual Basic



الأسئلة الموضوعية

أولاً: في البنود المرقمة من (١-٧) توجد عبارات صحيحة و أخرى غير صحيحة ، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة ، (ب) إذا كانت العبارة غير صحيحة :

١	للتخطيط لإنشاء برنامج نحدد البيانات التي سيتم إدخالها للبرنامج فقط .	أ	ب
٢	جميع البرامج تحتاج إلى مدخلات من المستخدم .	أ	ب
٣	حاوية الحل Solution يمكن أن تحتوي على أكثر من مشروع لموضوع واحد .	أ	ب
٤	تستخدم الأداة  لإنشاء مشروع جديد .	أ	ب
٥	تظهر أزرار التحكم في النافذة  تلقائياً في شريط العنوان .	أ	ب
٦	عند إغلاق المشروع يتم حفظ التعديلات تلقائياً .	أ	ب
٧	يمكن إضافة الكائنات من صندوق الأدوات Toolbox بالسحب والإفلات بالفأرة .	أ	ب

ثانياً: في البنود المرقمة من (١-٦) توجد قائمتان (الأولى والثانية)، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

عند إغلاق مشروع يظهر لنا صندوق حوار ويمكن منه إختيار:

الإجابة	القائمة الأولى (الزر)	القائمة الثانية (الوظيفة)
١	Save	أ) لفتح آخر مشروع تم العمل به .
٢	Discard	ب) لعدم إغلاق المشروع .
٣	Cancel	ج) لإغلاق المشروع وعدم حفظ التعديلات .
		د) لإغلاق المشروع مع حفظ التعديلات .

الفصل الأول : بداية بناء برنامج

يمكن استخدام أكثر من طريقة لإضافة الكائنات من صندوق الأدوات Toolbox

الإجابة	القائمة الأولى الطريقة	القائمة الثانية الوظيفة
٤	الضغط المزدوج على الأداة	تم إضافة العنصر بالحجم الذي تم رسمه في المكان الذي تم الرسم فيه .
٥	سحب الأداة وإفلاتها على النموذج	يضيف العنصر بحجمه الافتراضي أعلى يسار النموذج في حالة تفعيله أو فوق العنصر الفعال حالياً .
٦	اختيار الأداة ثم رسم الأداة على النموذج	يتم إضافة العنصر بحجمه الافتراضي في المكان الذي تم فيه تحرير زر الفأرة .
		يظهر صندوق حوار لتحديد موضع العنصر .

ثالثاً: في البنود المرقمة من (١-٣) لكل بند أربعة اختيارات واحد فقط منها صحيح ، اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها :

١	لإنشاء مشروع جديد من خلال البيسك المرئي ٢٠٠٨ يستخدم الأمر :		
	Save All	ب	Save Project
	Discard	د	New project
٢	لإضافة كائن من صندوق الأدوات Toolbox إلى النموذج والتحكم في موضعه وحجمه يتم :		
	اختيار الأداة ثم رسم الأداة على النموذج	ب	سحب الأداة وإفلاتها على النموذج
	الضغط المزدوج على الأداة	د	ليس أي مما سبق صحيح
٣	يستخدم الكائن Button لإضافة :		
	إطار يظهر بداخله صورة	ب	زر يرتبط بتفعيل تعليمات برمجية عند الضغط عليه بالفأرة
	إطار يظهر بداخله عنوان	د	صندوق نص يدخل المستخدم بيانات من خلاله

الفصل الأول : بداية بناء برنامج

رابعاً: في البنود المرقمة من (١-٣) عبارات تحتوي كل منها على فراغ ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة :

١	العمليات والتعليمات البرمجية التي ستتم على البيانات تسمى
٢	المعلومات المطلوبة الحصول عليها من ناتج المعالجة تسمى
٣	لحفظ جميع ملفات المشروع يستخدم الأمر

الأسئلة المقالية

أجب عن الأسئلة الآتية:

١- من خلال دراستك للتخطيط للبرنامج ، اذكر المقصود بكل من :

- (١) المدخلات :
- (٢) المعالجة:
- (٣) المخرجات:

٢- من خلال دراستك لإضافة الكائنات إلى النموذج أكمل الجدول الآتي بما يناسبه :

الوظيفة	الطريقة
يضيف العنصر بحجمه الافتراضي أعلى يسار النموذج في حالة تفعيله أو فوق العنصر الفعال حالياً .	الضغط المزدوج على الأداة المطلوبة
.....	سحب الأداة وإفلاتها على النموذج
يتم إضافة العنصر بالحجم الذي تم رسمه والمكان الذي تم الرسم فيه

الأسئلة الموضوعية

أولاً: في البنود المرقمة من (١-٧) توجد عبارات صحيحة وأخرى غير صحيحة ، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة ، (ب) إذا كانت العبارة غير صحيحة :

١	لا يمكن تغيير خصائص الكائن أثناء تشغيل البرنامج .	أ	ب
٢	عند تحديد الكائن يظهر اسمه في أعلى نافذة الخصائص .	أ	ب
٣	عند تحديد الكائن تتغير الخصائص إلى الخصائص المناسبة للكائن .	أ	ب
٤	علامة + بجانب الخاصية تعني أن لها خصائص فرعية .	أ	ب
٥	خاصية Text تحدد النص الذي سيعرض داخل الكائن .	أ	ب
٦	ليس هناك فرق بين خاصية Text وخاصية Name .	أ	ب
٧	يمكن ضبط خاصية Font لأكثر من كائن دفعة واحدة .	أ	ب

ثانياً: في البنود المرقمة من (١-٦) توجد قائمتان (الأولى والثانية)، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

الإجابة	اسم الخاصية الأولى	القائمة الثانية الوظيفة
١	خاصية Text .	تستخدم لتغيير لون الخط لنص الكائن .
٢	خاصية Font .	تحدد النص الذي سيعرض داخل الكائن .
٣	خاصية Anchor .	تحدد نوع وحجم ونمط الخط لنص الكائن .
		تحدد محاذاة الكائن عند تغيير حجم النافذة أثناء التشغيل .

الفصل الثاني : ضبط عناصر البرنامج

الإجابة	القائمة الأولى قيمة خاصية نمط الحجم SizeMode لكائن الصورة	القائمة الثانية الوظيفة
٤	StretchImage	يُجعل الإطار يساوي حجم الصورة .
٥	AutoSize	يُجعل الصورة تساوي حجم الإطار .
٦	Zoom	يُجعل الصورة تساوي حجم الإطار مع المحافظة على تناسب الطول مع العرض .
		يُظهر جزء من الصورة يناسب حجم الإطار .

ثالثاً: في البنود المرقمة من (١-٢) لكل بند أربعة اختيارات ، واحد فقط منها صحيح ، اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها :

١	الخاصية التي تحدد النص الذي سيعرض داخل الكائن هي :
أ	.Text
ب	.Name
ج	.Font
د	.Anchor
٢	الخاصية التي تستخدم لتغيير لون الخط لنص الكائن :
أ	.Font
ب	.ForeColor
ج	.Image
د	.Text

رابعاً: في البنود المرقمة من (١-٣) عبارات تحتوي كل منها على فراغ ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة :

١	الخاصية التي تحدد نوع وحجم ونمط الخط للنص هي
٢	الخاصية التي تحدد محاذاة الكائن عند تغيير حجم النافذة أثناء التشغيل هي
٣	الخاصية التي تستخدم لتغيير لون الخط لنص الكائن المحدد هي

الأسئلة المقالية

أجب عن الأسئلة الآتية:

١- عرف كل مما يلي :

(١) الخاصية :

.....

(١) خاصية النص Text :

.....

(١) خاصية الصورة Image :

.....

الأسئلة الموضوعية

أولاً: في البنود المرقمة من (١ - ٩) توجد عبارات صحيحة و أخرى غير صحيحة ، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة ، (ب) إذا كانت العبارة غير صحيحة :

١	تستخدم تعليمة End في إغلاق البرنامج .	أ	ب
٢	تحدد القاعدة التالية ObjectName.Properties=Vaule تفعيل طريقة لكائن	أ	ب
٣	التعليمة البرمجية Me.BackColor = Color.Yellow تعني لون خلفية للنموذج الحالي باللون الأصفر .	أ	ب
٤	تحدد القاعدة التالية ObjectName.Method تحديد قيمة لأحد خصائص الكائن	أ	ب
٥	يتجاوز البيسك المرئي عن أي خطأ إملائي في كتابة التعليمات البرمجية	أ	ب
٦	يجب كتابة تعليمات برمجية لكل حدث من أحداث الكائن .	أ	ب
٧	تظهر نافذة Output عمليات اختبار التعليمات البرمجية .	أ	ب
٨	المجلد الافتراضي لبناء البرنامج هو bin\Release .	أ	ب
٩	Public Class هي نهاية التعليمات البرمجية في النموذج .	أ	ب

ثانياً: في البنود المرقمة من (١ - ٦) توجد قائمتان الأولى والثانية، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

الإجابة	القائمة الأولى يرتبط بالكائن	القائمة الثانية
١	خصائص	وظائف مضمنه في الكائن .
٢	طرق	تحدد مظهره .
٣	أحداث	تشغيل البرنامج .
		مؤثرات على الكائن مثل الضغط بالفأرة .

الفصل الثالث : كتابة التعليمات البرمجية

الإجابة	القائمة الأولى يساعد أمر :	القائمة الثانية على :
٤	Save All	اختبار البرنامج (أ)
٥	Build	حفظ المشروع (ب)
٦	Start Debugging	بناء البرنامج (ج)
		حذف البرنامج (د)

ثالثاً: في البنود المرقمة من (١-٥) لكل بند أربعة اختيارات واحد فقط منها صحيح ، اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها :

١	تساعد الطريقة Hide () على :	
(أ)	إظهار الكائن .	(ب) إخفاء الكائن .
(ج)	تحريك الكائن .	(د) حذف الكائن .
٢	تساعد القيمة TimeString على :	
(أ)	إظهار الوقت الحالي	(ب) إظهار التاريخ الحالي
(ج)	إظهار وقت تصميم البرنامج	(د) ليس أيّاً مما سبق
٣	الحدث التلقائي للنموذج هو :	
(أ)	Click	(ب) Double Click
(ج)	Load	(د) ليس أيّاً مما سبق
٤	الحدث التلقائي للزر Button هو :	
(أ)	Click	(ب) Double Click
(ج)	Load	(د) ليس أيّاً مما سبق
٥	تساعد أداة  على :	
(أ)	حفظ البرنامج	(ب) حذف البرنامج
(ج)	اختبار البرنامج	(د) إنهاء البرنامج

رابعاً: في البنود المرقمة من (١-٣) عبارات تحتوي كل منها على فراغ ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة :

١	مع كتابة أول حرف من تعليمة برمجية في محرر التعليمات البرمجية تظهر
٢	لاختبار البرنامج نضغط على مفتاح
٣	يتم بناء البرنامج من خلال أمر

الأسئلة المقالية

أجب عن الأسئلة الآتية:

١- عرف كلاً من :

(أ) الحدث :

.....

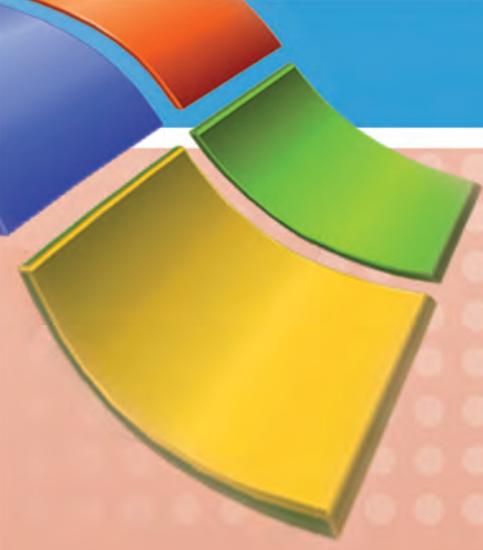
(ب) محرر التعليمات البرمجية :

.....

٢ - أكمل على الرسم بيانات الشكل التالي :





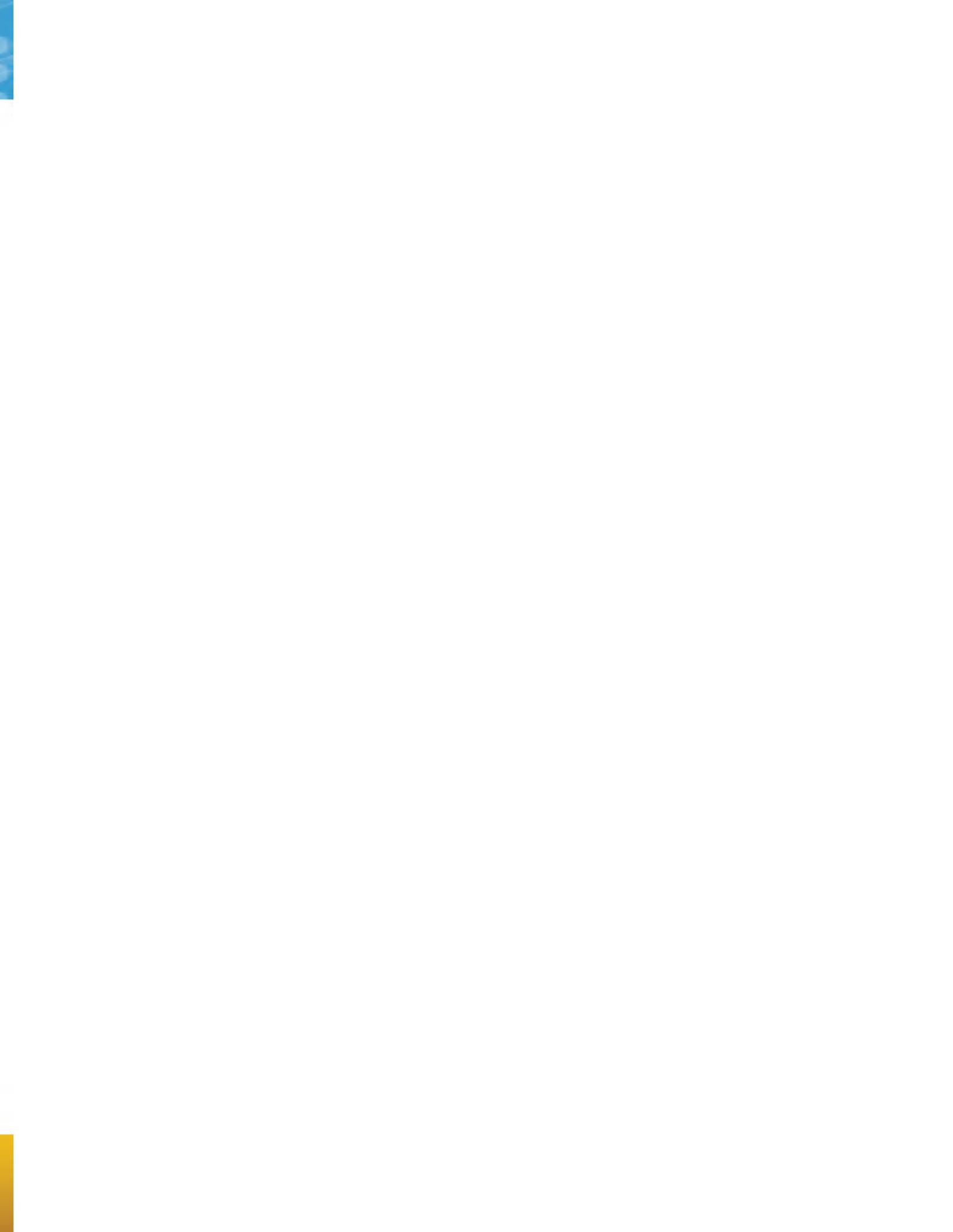


الباب الثالث

تعليمات برمجية أساسية



visual Basic



الأسئلة الموضوعية

أولاً: في البنود المرقمة من (١-١٧) توجد عبارات صحيحة وأخرى غير صحيحة، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة، (ب) إذا كانت العبارة غير صحيحة:

١	يمكن تخزين عدد أيام الغياب لمتعلم ما في متغير له النوع Date .	أ	ب
٢	القيمة الافتراضية لمتغير من النوع المنطقي «Boolean» هي True .	أ	ب
٣	جميع المتغيرات العددية لها القيمة الافتراضية (٠) صفر .	أ	ب
٤	عند الإعلان عن متغير وعدم تخصيص قيمة له تخصص له قيمة افتراضية حسب نوع المتغير .	أ	ب
٥	كلمة Dim تصلح كاسم للمتغير .	أ	ب
٦	يفرق البرنامج بين الأحرف الكبيرة والصغيرة في أسماء المتغيرات .	أ	ب
٧	يمكن الإعلان عن أكثر من متغير في جملة Dim واحدة .	أ	ب
٨	عند تخصيص قيم نصية لمتغير أثناء كتابة التعليمات البرمجية يجب وضعها بين علامتي # # .	أ	ب
٩	عند تخصيص قيم عددية لمتغير أثناء كتابة التعليمات البرمجية يجب وضعها بين علامتي « » .	أ	ب
١٠	عند الإعلان عن متغير على مستوى النموذج يمكن تخصيص قيمة له من أي مكان داخل النموذج .	أ	ب
١١	عند الإعلان عن متغير على مستوى المشروع يمكن تخصيص قيمة له من أي نموذج	أ	ب
١٢	الكلمة المفتاحية Me تدل على النموذج الحالي .	أ	ب
١٣	تستخدم الطريقة wohS لإخفاء النموذج .	أ	ب
١٤	تستخدم الدالة MsgBox لإظهار رسالة للمستخدم .	أ	ب
١٥	للإعلان عن ثابت نستخدم الكلمة المفتاحية eM .	أ	ب
١٦	يمكن الإعلان عن الثوابت بدون تحديد نوع لها .	أ	ب
١٧	قيمة الثابت تتغير أثناء تنفيذ البرنامج .	أ	ب

الفصل الأول : الثوابت والمتغيرات

ثانياً: في البنود المرقمة من (١ - ٩) توجد قائمتان الأولى والثانية، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

القائمة الثانية نستخدم متغيراً من نوع		القائمة الأولى لتخزين البيانات		الإجابة
Boolean	أ	النصية	١	
String	ب	التاريخ أو الوقت	٢	
Byte	ج	القيم المنطقية (True - False)	٣	
Date	د			

القائمة الثانية نستخدم متغيراً من نوع		القائمة الأولى لتخزين		الإجابة
Boolean	أ	اسم المتعلم	٤	
String	ب	تاريخ الميلاد	٥	
Date	ج	عدد أيام الغياب لمتعلم	٦	
Byte.	د			

القائمة الثانية نستخدم لـ		القائمة الأولى الكلمة		الإجابة
للإعلان عن متغير على مستوى النموذج أو الأجراء.	أ	Me	٧	
للإعلان عن متغير على مستوى المشروع.	ب	Dim	٨	
تخصيص قيم للمتغيرات .	ج	Public	٩	
للدلالة على اسم النموذج الحالي .	د			

ثالثاً: في البنود المرقمة من (١-٧) لكل بند أربعة اختيارات ، واحد فقط منها صحيح اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها :

١ القيمة الافتراضية لمتغير من النوع النصي هي :			
أ	صفر .	ب	سلسلة فارغة Null .
ج	2001/1/1 .	د	False .
٢ لتخزين عنوان المتعلم نستخدم متغيراً من نوع :			
أ	String .	ب	Date .
ج	Byte .	د	Boolean .
٣ واحد من أنواع المتغيرات التالية لا يصلح لتخزين بيانات عديدة :			
أ	Byte .	ب	Double .
ج	Date .	د	Integer .
٤ للإعلان عن متغير على مستوى النموذج أو الإجراء نستخدم الكلمة المفتاحية :			
أ	Public .	ب	Dim .
ج	Byte .	د	Me .
٥ عند اسناد قيمة نصية لمتغير من نافذة التعليمات البرمجية نستخدم :			
أ	»» .	ب	## .
ج	بدون أي علامات .	د	لا يمكن إسناد قيم للمتغيرات أثناء كتابة التعليمات .
٦ لإظهار رسالة للمستخدم نستخدم :			
أ	Hide .	ب	Me .
ج	Msgbox .	د	Show .
٧ لإخفاء النموذج نستخدم :			
أ	Dim .	ب	Show .
ج	Public .	د	Hide .

رابعاً: في البنود المرقمة من (١-٦) عبارات تحتوي كل منها على فراغ ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة :

١	مكان في الذاكرة تخزن فيه بيانات أثناء تنفيذ البرنامج ، له اسم فريد يميزه هو
٢	المتغير من نوع Boolean يستخدم لتخزين قيم
٣	المتغير من نوع Date يستخدم لتخزين قيم
٤	جميع المتغيرات العددية لها القيمة الافتراضية
٥	للإعلان عن متغير على مستوى المشروع نستخدم الكلمة المفتاحية
٦	للإعلان عن الثوابت داخل البرنامج نستخدم الكلمة المفتاحية

الأسئلة المقالية

أجب عن الأسئلة الآتية:

١- عرف كلاً من :

(أ) المتغير :

.....

(ب) نطاق المتغير :

.....

(ج) الوحدة النمطية Module :

.....

(د) الثابت :

.....

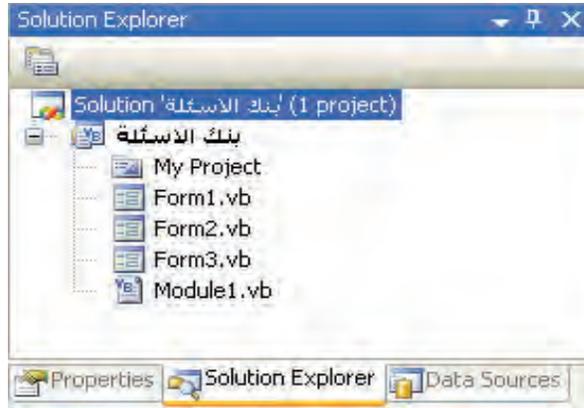
٢- اذكر ثلاثة من أنواع المتغيرات تستخدم لتخزين قيم عددية :

(أ)

(ب)

(ج)

٣- الشكل التالي يمثل نافذة مستكشف الحل ادرسه جيدا ثم أكمل الجدول التالي:



	اسم المشروع .
	عدد نماذج المشروع .
	اسم احدى نماذج المشروع .
	عدد الوحدات النمطية في المشروع .

٤- الشكل التالي يمثل نافذة التعليمات البرمجية ادرسه جيدا ثم أكمل الجدول التالي:

```

Public Class Form2
    Dim arb, eng, math As Double
    Dim s_name As String, s_date As Date
End Class
    
```

	عدد المتغيرات العددية .
	اسم احد المتغيرات العددية .
	اسم متغير يستخدم لتخزين حروف .
	اسم متغير لتخزين تاريخ .
	اسم النموذج الحالي .
	تم الإعلان عن المتغيرات على مستوى .

الأسئلة الموضوعية

أولاً: في البنود المرقمة من (١-٦) توجد عبارات صحيحة و أخرى غير صحيحة، ظلل في مكان الإجابة (أ) إذا كانت العبارة صحيحة، (ب) إذا كانت العبارة خاطئة:

١	عند استخدام تعليمة If Then فإن البرنامج ينفذ التعليمات بتسلسل مهما كان الشرط .	أ	ب
٢	عند استخدام تعليمة If Then يمكن كتابة أكثر من تعليمة برمجية ينفذها البرنامج إذا تحقق الشرط .	أ	ب
٣	تستخدم التعليمة End Select لإنهاء تعليمة If Then .	أ	ب
٤	تستخدم تعليمة Select case لتنفيذ عدة تعليمات برمجية حسب قيمة متغير معين .	أ	ب
٥	عند استخدام تعليمة Select case فإنه يمكن اختبار قيمة متغير تقع بين قيمتين .	أ	ب
٦	يمكن استخدام معاملات المقارنة داخل تعليمة Select case .	أ	ب

ثانياً: في البنود المرقمة من (١-٣) توجد قائمتان (الأولى والثانية، اختر لكل بند من القائمة الأولى ما يناسبه من القائمة الثانية، ثم اكتب بجوار رقم البند الحرف الذي يدل عليه:

الإجابة	عند استخدام التعليمة Select case فإن قيمة المتغير	القائمة الأولى	القائمة الثانية تعني عندما تكون قيمة المتغير
١	Case 10	أ	أقل من ١٠ .
٢	Case 8 to 10	ب	٨ أو ٩ أو ١٠
٣	Case 8.10	ج	٨ أو ١٠
		د	تساوي ١٠

الفصل الثاني : التفرع

ثالثاً: في البنود المرقمة من (١-٣) لكل بند أربعة اختيارات ، واحد فقط منها صحيح ، اختر الإجابة الصحيحة ثم ظلل دائرة الحرف الدال عليها :

١ لإنهاء تعليمة If Then تستخدم التعليمة :			
End Select	ب	End Case	أ
End Sub	د	End If	ج
٢ لإنهاء تعليمة Select case تستخدم التعليمة :			
End Select	ب	End Case	أ
Close	د	End If	ج
٣ عند استخدام تعليمة Select case لمتغير فإن التعليمة case is < 4 تعني أن يكون المتغير:			
أكبر من ٤	ب	١ أو ٢ أو ٣ أو ٤	أ
أقل من ٤	د	لا يساوي ٤	ج

رابعاً: في البنود المرقمة من (١-٣) عبارات، تحتوي كل منها على فراغ، أكمل الفراغ بما يناسبه من كلمات لتحصل على عبارات صحيحة:

١	لتنفيذ تعليمة برمجية إذا تحقق شرط واحد تستخدم التعليمة
٢	لتنفيذ عدة تعليمات برمجية حسب قيمة متغير ما تستخدم التعليمة
٣	لإنهاء تعليمة If Then تستخدم التعليمة

الأسئلة المقالية

ادرس التعليمات البرمجية الموضحة في الشكل الآتي ثم أكمل الجدول الذي يليه:

```
Public Class Form1
```

```
Dim x As Integer
```

```
Private Sub Button1_Click(ByVal sender As Object, ByVal e As System.  
EventArgs) Handles Button1.Click
```

```
    x = TextBox1.Text
```

```
    Select Case x
```

```
        Case Is > 10
```

```
            MsgBox(«تأكد من القيمة المدخلة»)
```

```
        Case 1 To 4
```

```
            MsgBox(«مستوى ضعيف»)
```

```
        Case 5
```

```
            MsgBox(«مستوى متوسط»)
```

```
        Case 6 To 10
```

```
            MsgBox(«مستوى متقدم»)
```

```
    End Select
```

```
End Sub
```

```
End Class
```

	اسم المتغير
	نطاق المتغير
	نوع المتغير
	قيمة المتغير
	ماذا سيحدث عندما تكون قيمة المتغير ١١
	ماذا يحدث عندما تكون قيمة المتغير ٥
	ماذا يحدث عندما تكون قيمة المتغير ٧

- * عزب محمد عزب، موسوعة مبرمجي فيجول بيسك ١٠٥، دار الكتب العلمية للنشر والتوزيع، مصر، 2007 م.
- * مايكل هالفرسون - مركز التعريب والبرمجة، خطوة خطوة Microsoft Visual Basic 2008 دار العلوم العربية، لبنان، 2008 م .
- * محمد حسين بصبوص وحمزة الغولة وخلدون الجدوع، البرمجة بلغة فيجوال بيسك للجميع، دار اليازوري العلمية، 2007 م.
- * تقنية المعلومات للصف الثاني عشر - ج ١ الطبعة الأولى 2008 / 2009م.
- * Christian Gross, Beginning VB 2008 From Novice to Professional, Apress, 2008.
- * James Foxall, Sams Teach Yourself Visual Basic 2008 in 24 Hours, Sams Press, 2008 .
- * Evangelos Petroustos , Mastering Microsoft Visual basic 2008, Wiley Publishing, 2008



