

Desktop

فى

مراجعة ليلة الامتحان

Microsoft
Visual Studio

الترم الثانى

Visual Basic

للمصف الثالث الإعدادى

إعداد

أ/ أحمد الشهاوى

الفصل الأول : البيانات Data

- يتم تخزين القيم في ذاكرة الكمبيوتر (Ram) بتمثيل معين يختلف حسب نوع هذه القيم.
- كل نوع بيان له طريقة تخزين في ذاكرة الكمبيوتر (Ram) وهذا يرجع في الأساس إلى استخدام ذاكرة الكمبيوتر الاستخدام الأمثل من حيث المساحة التخزينية، و العمليات الحسابية و المنطقية التي سوف تجرى على هذه القيم.
- تتميز لغة VB.NET بالتعامل مع أنواع مختلفة من البيانات.
- من أكثر أنواع البيانات استخداماً النوع String و يستخدم للبيانات النصية.
- أنواع البيانات (Byte - Short - Long - Integer) تستخدم للبيانات الرقمية الصحيحة.
- أنواع البيانات (Single - Double - Decimal) تستخدم للبيانات الرقمية الغير صحيحة (الكسور أو العشرية).
- تستخدم أنواع البيانات (Single - Double - Decimal) لتخزين البيانات الرقمية الصحيحة و الغير صحيحة .
- الخاصية Autosize نوعها بيانات متنوعة Boolean .
- يُصنف قيمة مجموع درجات الطالب ضمن البيانات الرقمية الغير صحيحة.
- يُصنف قيمة اسم الطالب ضمن البيانات الحرفية String.
- يُصنف قيمة نوع الطالب " ذكر " أم " أنثى " ضمن البيانات المتنوعة المنطقية Boolean .
- صورة الطالب يمكن تصنيفها ضمن البيانات المتنوعة Object.
- أنواع البيانات Boolean تستخدم للبيانات المنطقية و يأخذ True أو False.
- كل بيان يُخزن في ذاكرة الكمبيوتر يشغل مساحة تخزينية ومدى معين حسب نوع البيان.
- نوع البيان يُحدد حيز التخزين الذي يشغله في ذاكرة الكمبيوتر و معرفة الحد الأدنى والأقصى لقيمه.
- نوع البيان المُخزن مؤقتاً في ذاكرة الكمبيوتر يحدد حيز تخزيني ومدى قيمته.
- نوع البيان Date يستخدم لبيانات التاريخ و الوقت.
- قيمة مرتب الموظف يمكن تصنيفها ضمن البيانات الرقمية الغير صحيحة.
- كل نوع بيان (Data Type) له حيز تخزين في ذاكرة الكمبيوتر مثل (Integer) عند استخدامه يشغل 4 بايت.
- كل نوع بيان (Data Type) له مدى (Range) معين من القيم مثل نوع البيان (Byte) حدود القيم المخزنة به تبدأ من صفر وتنتهي بـ 255.
- أي بيانات يتم التعامل معها في أي برنامج تكون مخزنة في ذاكرة الجهاز الداخلية Ram .
- المبرمج الجيد الذي يُحسن ترشيد المساحة التخزينية في ذاكرة الكمبيوتر Ram .
- قيمة أسعار الأدوات المكتبية يمكن تصنيفها كبيانات رقمية غير صحيحة.
- قيمة أسماء المواد الدراسية يمكن تصنيفها كبيانات حرفية String .
- يختلف كل نوع من أنواع البيانات عن الآخر في المساحة التي يحتلها في الذاكرة و القيم التي يمكن حفظها في هذا النوع.
- الخصائص (Properties) هي مكان تخزين البيانات و أن لها نوع (Type) أو نوع بيان (Data Type) .
- المتغيرات (Variables) هي عبارة عن أماكن محجوزة بذاكرة الكمبيوتر (RAM) ولها نوع بيان (Data Type) ويتم ذلك بالإعلان عنها ثم تخصيص قيمة لها حسب نوع البيان أثناء سير تنفيذ أوامر و تعليمات البرنامج.
- الإعلان عن المتغيرات (Variables): تستخدم جملة (Dim) ثم كتابة اسم للمتغير (Variable) ثم كلمة (As) ثم نوع البيان (Data Type).
- جملة الإعلان عن المتغيرات يتحدد فيها اسم المتغير و نوعه.
- جملة الإعلان عن متغير Dim X As String تعني الإعلان عن متغير اسمه X ونوعه حرفي String.
- لا بد من الإعلان عن المتغيرات قبل استعمالها في البرنامج.
- المتغيرات يمكن تخصيص قيمة لها أثناء الإعلان عنها و أثناء سير (تشغيل) البرنامج (مرحلة لاحقة).
- استخدام كلمة Dim للإعلان عن المتغير و يجب ألا يحتوي السطر الواحد على أكثر من كلمة Dim.
- يمكن الإعلان عن أكثر من متغير في سطر واحد و يفصل بينهم بالفاصلة (,).
- عند الإعلان عن المتغيرات يمكن أن تكتب (=) و بعدها قيمة و هذه القيمة ستكون قيمة البداية ولا بد أن تكون من نفس نوع المتغير.

- عند الاعلان عن المتغيرات يمكن أن تهمل علامة (=) ولا تستخدم وفى هذه الحالة المتغيرات الرقمية تساوى صفر بينما المتغيرات الحرفية ستكون سلسلة حرفية فارغة.
- يجب أن يبدأ اسم المتغير أو الثابت بحرف من الحروف الانجليزية A-Z أو a-z أو علامة Underscore (_).
- اسم المتغير أو الثابت لا يحتوى على الرموز و العلامات الخاصة مثل (? , * , ^ , - , + , .) و غيرها.
- اسم المتغير أو الثابت يمكن أن يتكون من حروف و أرقام و علامة Underscore (_).
- لا تستخدم الكلمات المحجوزة لدى لغة (Visual Basic .NET) فى تسمية الثوابت و المتغيرات مثل (Dim, Single, as).
- يفضل أن يعبر اسم المتغير أو الثابت عن محتواه مثل المتغير (Price) لتخزين السعر.
- يجب اختيار نوع البيان المناسب لكل من الثوابت و المتغيرات بحيث تتناسب مع طبيعة البيانات التى سيتم تخزينها فيه.
- عند اختيار نوع البيان (Data Type) يراعى حجم البيانات التى سوف تخزن بالمتغير (Variable).
- عند اختيار نوع البيان (Data Type) يراعى تحديد نوع البيان (Data Type) القيم التى سوف تخزن بالمتغير (Variable) بناء على العمليات الحسابية والمنطقية التى من المتوقع أن تجرى على المتغير (Variable).
- عند اختيار نوع البيان (Data Type) يراعى حجم الذاكرة الذى يحتاجها البيان.
- جملة التخصيص (Assignment): عبارة عن جملة من طرفين بينهما علامة التخصيص (=) ولا يقصد بها التساوى الحسابى حيث الطرف الأيسر المكان الذى يوضع فيه القيمة أو المتغير أو الخاصية و الطرف الأيمن عبارة عن قيمة الخاصية أو المتغير.
- يمكن تخصيص القيم للمتغيرات (Variables) أثناء الاعلان عنها أو فى مرحلة لاحقة.
- يمكن إعادة التخصيص للمتغيرات أى عدد من المرات و لذلك أطلق لفظ المتغيرات لأن قيمتها يمكن أن تتغير أثناء سير عمليات البرنامج فى أى وقت.
- جملة التخصيص تستخدم لتخزين قيمة مجردة أو متغير و خاصية أو تعبير.
- تخصيص قيمة مجردة مثل Number=5 .
- تخصيص قيمة من متغير آخر مثل Number2 = Number1 .
- تخصيص قيمة من خاصية مثل Name = Label1.Text .
- تخصيص قيمة من تعبير Expression مثل Number = Number/3 .
- تخصيص قيمة Value مثل Dim Number As Single=5.6 .
- يفصل بين كل متغير و الآخر (&VBCRLF&).
- المتغيرات تكتب فى جمل التخصيص سواء على يمين أو يسار معامل التخصيص.
- علامة الربط & تستخدم للربط بين النصوص.
- الكلمة المحجوزة VBCRLF (Visual Basic Carriage Return Line Feed) تستخدم فى إنشاء سطر جديد وهى تشير لمفتاح الإدخال Enter من لوحة المفاتيح.
- تستخدم علامة Underscore (_) حتى يمكن كتابة سطر الكود على أكثر من سطر فى حالة إذا كان سطر الكود (Code) طويل بعض الشئ وذلك للتنظيم و تسهيل عملية قراءة الكود (Code).
- (Me) تعبر عن نافذة النموذج (Form) الحالية.
- يستخدم المبرمج الامر Rem أو العلامة (') فى كتابة ملاحظات يمكن الرجوع إليها داخل الكود ولا يتم ترجمتها.
- ما يكتب بعد Rem أو العلامة (') لا يعتبر أكواد و يهمل عند ترجمة الأكواد (Code) بواسطة مترجم (Compiler) اللغة.
- الثوابت Constants هى عبارة عن أماكن محجوزة بذاكرة الكمبيوتر (RAM) ولها نوع بيان (Data Type) ويتم ذلك بالاعلان عنها ثم تخصيص قيمة لها حسب نوع البيان (Data Type) أثناء الاعلان عنها فقط.
- قيمة الثابت لا يمكن أن تتغير أثناء سير تنفيذ أوامر و تعليمات البرنامج.
- الاعلان عن الثوابت Constants: هى عملية إعطاء أسماء للخلايا إذا كانت الخلايا ستحمل قيم ثابتة.

- تستخدم جملة **(Const)** ثم كتابة اسم للثابت **Constants** ثم كلمة **(As)** ثم نوع البيان **(Data Type)** ثم علامة **(=)** ثم القيمة.
- يجب تخصيص قيمة للثوابت أثناء الإعلان عنها فقط.
- تستخدم علامتي التنصيص " " في حالة إذا ما أردنا كتابة نص.
- تستخدم علامتي الشباك # # في حالة إذا ما أردنا كتابة تاريخ أو وقت.
- لا بد من الاعلان عن الثوابت قبل استعمالها في البرنامج.
- لا بد من استخدام كلمة **Const** للاعلان عن الثابت وهي من الكلمات المحجوزة.
- عند الاعلان الثوابت لا بد أن تكتب **(=)** ثم بعدها قيمة هذا الثابت ولا بد أن تكون هذه القيمة من نفس نوع الثابت.
- يقصد بنطاق المتغيرات أو الثوابت موضع الاجراء الذي يتم فيه الاعلان عن الثابت أو المتغير واستخدامه.
- عند الاعلان عن الثابت الرياضي نستخدم الكود **Const Pi As Single=3.14**.
- عند الاعلان عن ثابت عجلة الجاذبية الأرضية نستخدم الكود **Const g As Single=9.81**.
- لا يمكن استخدام المتغيرات و الثوابت إلا في نطاق إعلانها.
- متغير أو ثابت محلي **Local** : هي التي يتم الاعلان عنه داخل الاجراء (حدود استخدامه هو مستوى الاجراء).
- متغير أو ثابت على مستوى التصنيف **Class** : هي التي يتم الاعلان عنها في تصنيف النموذج **Form** (حدود الاستعمال هو مستوى النموذج **Form**).
- عند كتابة الكود **Code** لا بد من الالتزام بقواعد لغة البرمجة المستخدمة.
- الأخطاء الإملائية والنحوية **Syntax Errors** أي أخطاء في قواعد اللغة وتحدث عند كتابة الكود **Code** بصورة غير سليمة.
- الأخطاء المنطقية **Logic Errors** وتحدث إذا استخدمت تعبيرات **Expressions** بناؤها غير سليم وتحدث في جمل التخصيص غالباً ، مما يؤدي إلى نتائج غير سليمة عند استخدام البرنامج ولا تعطى رسائل خطأ .
- الخطأ في نتيجة حساب أي معادلة يعتبر خطأ منطقي **Logical Error**.
- الخطأ الذي يظهر عند ترك صندوق النص فارغ دون ادخال البيانات يطلق عليه خطأ أثناء التشغيل **Runtime Error**.
- الخطأ الذي يظهر عند ادخال قيمة نصية (في متغير رقمي) بدلاً من رقمية يطلق عليه خطأ أثناء التشغيل **Runtime Error**.
- الخطأ الذي يظهر عند ادخال قيمة أكبر من الحد المسموح **Overflow** يطلق عليه خطأ أثناء التشغيل **Runtime Error**.
- الخطأ الذي يظهر أثناء تنفيذ برنامج **VB.NET** يطلق عليه خطأ أثناء التشغيل **Runtime Error**.
- الخطأ الذي يظهر أثناء كتابة كود بلغة **VB.NET** يسمى الأخطاء الإملائية والنحوية **Syntax Errors**.
- رسالة الخطأ التي تظهر عند كتابة الكود **Dimension X As Byte** يمكن تصنيفها خطأ الأخطاء الإملائية والنحوية **Syntax Errors**.
- أولويات العمليات الحسابية :
 1. فك الأقواس من الداخل إلى الخارج () () () .
 2. الرفع إلى الأس ^ .
 3. الضرب * والقسمة العادية / أيهما أسبق من اليسار إلى اليمين.
 4. قسمة الأعداد الصحيحة \ (صحيح القسمة) .
 5. باقى القسمة Mod .
 6. الجمع + والطرح - أيهما أسبق من اليسار إلى اليمين.
- من الضروري الالتزام بمراعاة أولويات تنفيذ العمليات الحسابية عند إنشاء تعبيرات **Expression** حسابية تستخدم في جملة التخصيص **Assignment** حتى لا تقع أخطاء من النوع **Logic Errors** لأن ناتج جملة التخصيص سيتغير في هذه الحالة دون ظهور أي رسائل خطأ.



المراجعة النهائية لا يخرج عنها الامتحان بإذن الله

- يتم احتساب قيمة التعبيرات الحسابية من المستوى الأعلى في الأسبقية إلى المستوى الأقل (أولويات العمليات الحسابية).
- في حالة وجود أكثر من عملية لها نفس الأولوية كالضرب والقسمة - أو الجمع والطرح - فإن ترتيب تنفيذها يكون من اليسار إلى اليمين .
- إذا تداخلت الأقواس فإن الأقواس الداخلية يتم تنفيذ ما بينها أولاً.
- ناتج التعبير الحسابي دائماً ما يكون قيمة عددية.
- ناتج التعبير الشرطي دائماً ما يكون True أو False.

الفصل الثاني : التفرع Branching

- التفرع باستخدام جملة `If ..Then ...` : وهي من أوامر التفرع لاختيار بديل واحد فقط أو مجموعة واحدة من الأوامر (أحادية الاختيار أو Single Selection).
- التفرع باستخدام جملة `If ..Then ...` إذا كان ناتج الشرط صواب True يتم تنفيذ الاختيار `Code1`.
- التفرع باستخدام جملة `If ..Then ...` إذا كان ناتج الشرط خطأ False يخرج من جملة الشرط.
- عدد اختيارات التفرع الممكنة مع جملة (If .. Then) هو 1 (واحد) .
- التفرع باستخدام جملة `If ... Then ... Else ...` : وهي من أوامر التفرع لاختيار بديل واحد من بين بديلين أو مجموعة من مجموعتين من الأوامر (ثنائية الاختيار أو Double Selection).
- التفرع باستخدام جملة `If ... Then Else ...` إذا كان ناتج الشرط صواب True يتم تنفيذ الاختيار الأول `Code 1`.
- التفرع باستخدام جملة `If ... Then ... Else` إذا كان ناتج الشرط خطأ False يتم تنفيذ الاختيار الثاني `Code 2`.
- التفرع باستخدام جملة `If ... Then ... Else ...` تنتهي جملة If الشرطية بـ `End If`.
- التفرع باستخدام جملة `If ... Then ... Else ...` آخر Else تعتبر شرطاً افتراضياً فإذا لم يوجد أي ناتج صواب True فينفذ مجموعة الأوامر الخاصة بآخر Else.
- عدد اختيارات التفرع الممكنة مع جملة (If .. Then .. Else) هو 2 (اثنين) .
- لإفراغ محتوى الأداة `Textbox` يستخدم الكود التالي " `Textbox1.text=` " .
- كلمة `Else` تعني تنفيذ ما بعدها إذا كان ناتج الشرط خطأ False.
- إذا كان ناتج التعبير الشرطي صحيحاً True سيتم تنفيذ الأوامر التي تلي كلمة `Then`.
- أما إذا كان ناتج التعبير الشرطي خطأ False سيتم تنفيذ الأوامر التي تلي كلمة `Else`.
- يمكن كتابة الكود `Code` لجملة `If then` في سطر واحد وعندها لا يتم كتابة أمر `End If`.
- التعبيرات الشرطية `Conditional Expression` : هو جزء من كود `Code` البرمجة يكون ناتجه دائماً إما صواب True أو خطأ False وذلك بناء على قيمة خاصية `Property` أو متغير `Variable` أو بيان آخر بالبرنامج (قيمة مجردة) .

معاملات المقارنة :

المعنى	المعامل	المعنى	المعامل
أكبر من	>	أصغر من أو يساوي	< =
أصغر من	<	يساوي	=
أكبر من أو يساوي	> =	لا يساوي	< >

- المعامل `Mod` هو معامل حسابي يعطي باقي قسمة عددين .
- لتشغل البرنامج من أيقونة `Start Debugging` ▶ من شريط الأدوات القياسي أو بالضغط على مفتاح `F5`.
- التفرع باستخدام جملة `Select Case` : فعالة بشكل أكثر عندما يكون التفرع معتمداً على قيمة متغير `Variable` واحد فقط أو خاصية واحدة فقط ، وهي تستخدم في حالة وجود احتمال للتفرع أو أكثر.
- تستخدم دالة `Focus()` وهي تعني نقل التركيز و وضع مؤشر الكتابة بداخل صندوق النص `TextBox`.

- أداة **ListBox** تتيح لك أن تنشأ و تعرض قائمة من العناصر يمكن لمستخدم البرنامج أن يختار منها عنصر أو أكثر.
- **Index** تشير إلى رقم العنصر داخل الأداة.
- ترتيب **Index** العناصر داخل الأداة **ListBox** يبدأ من الرقم صفر للعنصر الأول و العنصر الثاني يأخذ الرقم 1 و العنصر الثالث يأخذ الرقم 2 و هكذا .
- الحدث الافتراضي للأداة **ListBox** هو **SelectedIndexChanged** عند تحديد المستخدم عنصر آخر بدلاً من المحدد الحالي.
- **SelectedIndex** الخاصية التي تشير إلى العنصر المحدد بالأداة **ListBox** .

الفصل الثالث : الحلقات التكرارية والإجراءات Loops And Procedures

- استخدام الجملة **For Next** : يستخدم لتكرار تنفيذ كود **Code** لعدد محدد من المرات معروف مسبقاً عددها.
- جملة **For Next** هي أحد جمل التكرار المحدود.
- يبدأ الإعلان عن الحلقة التكرارية بكلمة **For** وتنتهي بكلمة **Next** .
- يستمر التكرار في جملة **For Next** طالما قيمة العداد لم تتخطى قيمة نهاية العداد **End** .
- اسم المتغير **Variable** يطلق عليه متغير عداد **Counter** الحلقة نوع بيانه عددي (صحيح أو غير صحيح) .
- إذا كانت قيمة زيادة العداد موجبة (1) ، فإنه يمكن الاستغناء عن كتابة **Step Add Value** باعتبار أن القيمة الافتراضية لزيادة العداد موجب (1) .
- يمكن تحديد معدل (قيمة) الزيادة للمتغير أو العداد **Counter** بكلمة **Step** ثم كتابة الرقم.
- كتابة اسم متغير **Variable** العداد بجوار **Next** اختيارية.
- وظيفة الأمر **Next** هي زيادة عداد الحلقة (المتغير) بقيمة معدل الزيادة.
- وظيفة الأمر **Next** هي مقارنة القيمة الجديدة لعداد الحلقة (المتغير) بقيمة النهاية **End** للحلقة التكرارية.
- في جملة **For ... Next** عند عدم تخطى القيمة الجديدة (نقصانها عن) قيمة النهاية يتم تكرار الكود **Code** أو الأوامر التي تلي الأمر **For**.
- في جملة **For ... Next** عندما تتخطى القيمة الجديدة (زيادتها عن) قيمة النهاية يتم الخروج من الحلقة وتنفيذ الكود الذي يلي الأمر **Next** وهو **End Sub**.
- يمكنك جعل قيمة البداية **Start** أكبر من قيمة النهاية **End** وفي هذه الحالة يجب أن تكون قيمة الزيادة **Step** سالبة.
- معدل (قيمة) الزيادة للمتغير أو العداد **Counter** يجب أن يكون سالباً في حالة أن تكون قيمة البداية أكبر من قيمة النهاية .
- يمكن أن تكون قيمة البداية **Start** و قيمة النهاية **End** و قيمة الزيادة **Step** رقم صحيح أو عشري أو متغير عددي أو ثابت عددي ، رقم سالب أو موجب.
- وضع بعض الكلمات بين قوسين بهذا الشكل [] وهذا للدلالة على أنها اختيارية أي يمكن عدم كتابتها وفي هذه الحالة تأخذ قيمة افتراضية هي (1+) .
- يمكن تحديد معدل (قيمة) الزيادة للمتغير أو العداد **Counter** بكلمة **Step** ثم كتابة الرقم.
- يمكن لقيمة البداية أو النهاية أو معدل الزيادة أن يكون عدد عشري وفي هذه الحالة يجب تعريف متغير الحلقة من نوع **Data Type** يقبل الكسور العشرية مثل النوع **Single**.
- يمكن لقيمة البداية أو النهاية أو معدل الزيادة أن يكون متغير **Variable** أو ثابت عددي .
- يجب أن يكون عداد الحلقة **Counter** متغير **Variable** عددي مناسب .
- تستخدم الجملة **Do While Loop** لتكرار تنفيذ كود (Code) محدد لعدد من المرات غير معروف مسبقاً بناءً على شرط معين .
- استخدام الجملة **Do While Loop** مفيدة في حالة عدم معرفة عدد مرات التكرار بشكل قاطع.
- المقصود بجملة **Do While** تكرار الكود **Code** طالما التعبير الشرطي صحيح **True**.
- أمر التكرار **Do** يستخدم لتكرار مجموعة من الأوامر إذا تحقق شرط معين (**True**) ولا يعرف لها عدد مرات تكرار محدد مسبقاً .

- ينتهي التكرار بجملة **Do While** عندما لا يتحقق الشرط (**False**) .
- الأمر **Loop** يدل على نهاية جملة التكرار.
- لإنهاء حلقة تكرارية عند تحقق الشرط **A=2** تكتب بدايتها **Do While A<>** (لإنهاء الحلقة التكرارية لا بد من عكس الشرط).
- لتلافى تكرار إضافة الكتابة داخل **ListBox1** عند تنفيذ أى من الإجراءات مرة أخرى يجب كتابة الكود التالي :
Me.ListBox1.Items.Clear()
- يستخدم الكود **Me.ListBox1.Items.Clear()** لمسح محتويات صندوق القائمة **ListBox1**.
- الإجراء **Procedure** : هو مجموعة من الأوامر والتعليمات **Code** تحت اسم ما ، عند استدعاء هذا الاسم يتم تنفيذ هذه الأوامر والتعليمات .
- الإجراء **Procedure** إما أن يكون إجراء **Sub** لا تعود بقيمة .
- الإجراء **Procedure** إما أن يكون دالة **Function** تعود بقيمة .
- نلجأ للإعلان عن الإجراء **Sub** فى حالة إذا ما كان لدينا كود **Code** سيتكرر كتابته فى أكثر من موضع داخل التصنيف **Class** .
- نلجأ للإعلان عن الإجراء **Sub** بهدف تنظيم كتابة الكود **Code** حتى يسهل قراءته وفهمه ومن ثم تعديله .
- يبدأ الإجراء بأمر **Sub** ثم اسم الإجراء ثم بين القوسين نحدد وسائط أو معاملات (**Parameters**) الإجراء ويلى ذلك الأوامر و التعليمات المطلوب تنفيذها ثم يلى ذلك عبارة **End Sub** تعبر عن نهاية الإجراء.
- فى الإجراء **Sub** يمكن استخدام أى عدد من الوسائط .
- يستدعى الإجراء بكتابة اسمه و وضع قوسين فى نهاية الاسم.
- الإجراءات **Sub** لا يجوز استخدامها فى أى جملة تخصيص .
- يتم الإعلان عن الإجراء **Sub** مرة واحدة و يمكن استدعاؤه أى عدد من المرات .
- يفضل تسمية الإجراء **Sub** و الدالة **Function** تسمية متعلقة بوظيفتها أو محتواها .
- نلجأ للإعلان عن دالة **Function** فى حالة إذا ما كان لدينا **Code** سينتج (تعود) منه قيمة نحتاجها .
- يبدأ الإعلان عن الدالة بأمر **Function** ثم اسم الدالة ثم بين قوسين نحدد المتغيرات التى تستخدم كوسائط (**Parameters**) للدالة عند استدعاؤها ثم نوع الدالة يلى ذلك الأوامر و التعليمات **Code** المطلوب تنفيذها ثم يلى ذلك تحديد القيمة الراجعة باستخدام أمر **Return** يلى ذلك عبارة **End Function** تعبر عن نهاية الدالة .
- الدوال **Functions** لا يمكن تخصيص قيم لها .
- يجب أن يكون قيم الوسيط **Argument** من نوع بيان متوافق مع نوع بيان الوسيط **Parameter** .
- يمكن الإعلان عن إجراء **Procedure** بدون استخدام أى معامل أو وسيط **Parameter** أو باستخدام معامل أو أكثر .
- تستخدم المعاملات أو الوسائط **Parameters** كوسيلة لاستقبال قيم **Argument** من خارج الإجراء .
- الدوال **Functions** يتم استدعاؤها فتنتج قيمة تخزن بها ومن ثم يمكن استخدام هذه القيمة المخزنة بها .
- الدوال **Functions** يفضل أن يكون الاسم متعلق بوظيفة الدالة أو محتواها ، ولها نوع بيان .
- يتم الإعلان عن الدالة مرة واحدة و يمكن استدعاؤها أى عدد من المرات .
- **Value** القيمة الراجعة (العائدة) من الدالة **Function** التى يجب الإعلان عنها داخل الدالة بنوع بيان يوافق نوع البيان الخاصة بالدالة **Function** (**Name**) .
- ليس من الضروري أن يكون نوع بيان **Data Type** للمعاملات (الوسائط) من نفس نوع بيان **Data Type** الدالة أو القيمة الراجعة .
- من الضروري أن يكون المتغيرات من نوع بيان يوافق نوع بيان وسائط أو معاملات **Parameters** الدالة **Function** .
- جميع الدوال **Function** تستخدم فى الطرف الأيمن فقط من معادلة التخصيص للحصول على نتائجها .

- الإجراءات **Sub** لا يجوز استخدامها في أي جملة تخصيص.
- إجراء الحدث **Event Procedure** يعتبر إجراء من النوع **Sub**.
- وسيط الإجراء **Procedure** إما أن يكون قيمة مجردة أو متغير **Variable** أو ثابت **Constant** أو دالة **Function**.
- الإجراء **Sub** لا يمكن تخصيص قيمة لها ، وإنما تستدعي باسمها لتنفيذ الأوامر التي بها .
- الإجراء **Sub** ليس له قيمة مخزنة في الذاكرة لأنه لا يعود بقيمة وليس لها ناتج.
- الإجراء **Sub** ليس له نوع بيان **Data Type** لأنه لا يعود بقيمة.
- الإجراء **Sub** لا يجوز أن تكتب في أي جمل تخصيص نهائياً.
- الدوال **Functions** لا يمكن تخصيص قيمة لها ، وإنما تستدعي باسمها فتنتج قيمة تخزن بها.
- الدوال **Functions** يمكن استخدام القيم المخزنة بها في الذاكرة و لها ناتج.
- الدوال **Functions** لها نوع بيان **Data Type** هو نفس نوع بيان القيمة الراجعة.
- الدوال **Functions** تكتب في جمل التخصيص على يمين معامل التخصيص فقط ، لأنها لا يخصص لها قيمة.
- الدوال **Functions** قد يكون لها وسائط ومعاملات قد تكون قيم مجردة أو متغيرات أو ثوابت.

Sub ShowOddOrEven (ByVal Start As Integer)

Dim i As Integer

Label1.Text = ""

For i = Start To 10 Step 2

Label1.Text = Me.Label1.Text & " " & i

Next

End Sub

م	المطلوب	الإجابة
1	اسم الإجراء	ShowOddOrEven
2	تم الإعلان عن Parameters باسم	Start
3	نوع Parameters	Integer عددي صحيح
4	الحلقة التكرارية تبدأ القيمة	Start
5	قيمة الزيادة في الحلقة التكرارية تساوي	2
6	عند استدعاء الإجراء لتنفيذ الكود بدءاً من القيمة	Start

Function XXX (ByVal YYY As Integer, ByVal ZZZ As Integer) As Single

Code

Return RRR

End Function

- 1- اسم الدالة **XXX**
- 2- نوع البيان الخاص بالقيمة الراجعة من الدالة **Single**
- 3- الوسائط **Parameter** التي سوف تستخدم في الكود **ZZZ & YYY** .
- 4- القيمة الراجعة من الدالة **RRR**.

الفصل الرابع : التعدي الإلكتروني

- التعدي الإلكتروني عبارة عن سلوك عدواني متعمد يستخدم الوسائط الإلكترونية بغرض التحرش - المضايقة - الإحراج - التخويف - التهديد الآخرين.
- الوسائط الإلكترونية للتعدي مثل البريد الإلكتروني E-Mail - المنتديات الإلكترونية Forums - الرسائل الفورية Instant Message - التدوين الإلكتروني Blogging - مواقع الانترنت - المواقع الاجتماعية مثل Facebook - الهاتف المحمول .
- من أشكال التعدي الإلكتروني مثل التخفي الإلكتروني Anonymity & المضايقات الإلكترونية Harassment & الملاحقة الإلكترونية Cyber stalking & السب أو القذف الإلكتروني Flaming & التشهير الإلكتروني Outing & الاستثناء الإلكتروني (Exclusion) & التهديد الإلكتروني (Threats Cyber) & التعدي من خلال النص و الدردشة (Chatting) & نشر صور غير لائقة.
- التخفي الإلكتروني Anonymity استخدام أسماء مستعارة تخفي شخصية المتعدي الإلكتروني بغرض الإفلات من العقاب.
- المضايقات الإلكترونية Harassment عبارة عن رسائل عدائية ومزعجة موجهة ضد شخص أو أكثر.
- الملاحقة الإلكترونية Cyber stalking هي إرسال تعليقات عدائية بشكل متكرر لشخص ما بهدف التهديد أو التخويف.
- السب أو القذف الإلكتروني Flaming عبارة عن نشر تعليقات أو كلمات عدائية و مبتذلة ضد شخص أو أكثر.
- التشهير الإلكتروني Outing عبارة عن نشر معلومات عن شخص محدد أو أكثر بشكل مسيء.
- الاستثناء الإلكتروني (Exclusion) عبارة عن ضم بعض المراهقين إلى مجموعة معينة و نبذ الآخرين و ألعاب الشبكة و يتجاهلون أحدهم كأنه لم يلعب معهم مع أنه موجود على الشبكة.
- التهديد الإلكتروني (Threats Cyber) عبارة عن إرسال رسائل إلكترونية تحمل تهديد أو وعيد لشخص أو أكثر .
- التعدي من خلال النص و الدردشة (Chatting) هو المحادثة عبر الانترنت باستخدام نصوص و عبارات غير مرغوب فيها.
- نشر صور غير لائقة هو ظهور بعض الصور غير اللائقة لبعض الأشخاص أثناء التجول عبر مواقع الانترنت.
- لا تشارك أحد بكلمة السر، مهما كان قريب منك (كصديق) ، وأن تتعامل مع كلمة السر كمفتاح شفتك لا تعطيه لأي صديق من أصدقائك.
- إعداد كلمة مرور لا تستنتج ، وذلك باستخدام مزيج من الحروف والأرقام " غير متعلقة باسمه أو سنة ميلاده أو تفضيلاته أو أبنائه أو الأشخاص المقربين جداً منه".
- عدم نشر أي بيانات خاصة أو عائلية على الإنترنت، بغض النظر عن طريق البريد الإلكتروني، أو النصوص المكتوبة، أو اليوميات الإلكترونية، أو على المواقع الإلكترونية.
- عدم حذف رسائل التعدي الإلكتروني ، حيث يمكن أن تحتاجها كأدلة ضده.
- عدم مقابلة أحدا تعرفت عليه من خلال الإنترنت.
- عدم إرسال رسائل وأنت في حالة غضب.
- اطلاع ولي الأمر بما يضايقك عند استخدام الإنترنت أو ممن تثق بهم من كبار العائلة أو من المعلمين إذا تلقيت بريد أو رسالة إلكترونية تجعلك تشعر بعدم الراحة ، و أخبر وشارك أحد الكبار بتجاربك على شبكة الإنترنت.
- إنزال البرامج من الإنترنت يكون تحت إشراف معلمك أو ولي أمرك.
- قم بالإبلاغ عن التعدي الإلكتروني للسلطات المختصة.
- لا ترد على المتعدي إلكترونياً، ولا تصدق كل ما يكتبه على الإنترنت والانتظار حتى تهدأ.
- احترم دائماً حق الآخرين على شبكة الإنترنت، وتعلم الخلق الحسن الذي يعتبر فن أدب الحوار على شبكة الإنترنت.

مع أطيب التمنيات بالنجاح والتوفيق