

** الفصل الأول **

البيانات Data

مقدمة:

تعلمنا فيما سبق كيفية إعداد واجهة البرنامج باستخدام لغة Visual Basic .NET (وكيفية ضبط الخصائص) Properties في وضع التصميم من خلال نافذة الخصائص Properties وتعرفنا على نافذة الكود Code window و إجراءات الأحداث Event Procedure وكيفية تخصيص قيم برمجية لخصائص بعض الأدوات Controls ومن خلالها اتضح لنا أنه يمكن استخدام البرنامج أن يدخل قيم (بيانات) ذات أنواع مختلفة كما هو موضح

نلاحظ: ان (ياسمين تامر، ٧-١٢-٢٠٠٥، انثى، ٥) عبارة عن قيم مختلفة في نوعها حيث:

اسم الحقل	القيمة	نوع القيمة
الاسم	ياسمين تامر	نصية
تاريخ الميلاد	٢٠٠٥-١٢-٧	تاريخ
النوع	انثى	منطقية
عدد افراد الأسرة	٥	رقمية

ملاحظات هامة

- ١- كل هذه القيم يتم تخزينها في ذاكرة الكمبيوتر RAM بتمثيل معين يختلف حسب نوع هذه القيمة
- ٢- وكل نوع من هذه البيانات له طريقة تخزين مختلفة في ذاكرة الكمبيوتر وذلك يرجع إلي الاستخدام الأمثل لذاكرة الكمبيوتر من حيث المساحة التخزينية و العمليات الحسابية والمنطقية التي ستجرى على هذه البيانات

سؤال

ما هو سبب اختلاف طريقة تخزين أنواع البيانات في ذاكرة الكمبيوتر؟

- ١- حتى يتم استخدام ذاكرة الكمبيوتر الاستخدام الأمثل من حيث المساحة التخزينية المخصصة لكل نوع بيان
- ٢- وكذلك من حيث العمليات الحسابية والمنطقية التي ستجرى على هذه البيانات او القيم

أنواع البيانات Data Type

توفر لغة Visual Basic .NET العديد من أنواع البيانات (Data Types) التي تستخدم في تخزين قيم البيانات التي يدخلها المستخدم أو الناتجة من تنفيذ الأوامر أو التعليمات الخاصة بالبرنامج

- ١- عندما يتم إدخال البيانات بواسطة وحدات الإدخال فإنها تخرن داخل الذاكرة الداخلية
- ٢- وتتكون ذاكرة الحاسب من ملايين الخلايا (الأمكان) المتساوية وتسمى هذه الخلايا بالبايت (byte) وتحمل هذه الخلايا أرقاماً متسلسلة تبدأ من الصفر
- ٣- تمثل عدد هذه الخلايا حجم ذاكرة الحاسب ((أي انه كلما زاد عدد هذه الخلايا (byte) كانت ذاكرة الجهاز اكبر وبالتالي فان سعة تخزين البيانات تكون اكبر))

أهم أنواع البيانات Data Type

أولاً : البيانات الرقمية (Numeric Data Types) : ونستخدم في تخزين الأرقام .

هي التي يمكن استخدامها في تخزين البيانات الرقمية وتنقسم إلى

١- البيانات الرقمية الصحيحة (Integral Numeric Types) : وهي التي يمكن أن تخزن بها أرقام (أعداد) صحيحة ومنها : (Long – Integer – Short – Byte)

٢- البيانات الرقمية الغير الصحيحة (No integral Numeric Types) : وهي التي يمكن أن تخزن بها أرقام بعلامات عشرية . ومنها : (Decimal – Double – Single)

ثانياً : البيانات الحرفية (Character Data Type) : ونستخدم في تخزين الحروف والكلمات ومنها : (Char – String)

ثالثاً : البيانات المتنوعة (Miscellaneous Data Types) : هي البيانات التي لا تندرج تحت تصنيف البيانات الرقمية أو الحرفية ونستخدم في تخزين بيانات منطقية أو تاريخ أو غيرها ومنها : (Boolean – Date – Object)

يتم تحديد نوع البيانات Data Types طبقاً لـ

١. حجم المتغير (عدد الـ Bytes) أي حجم الذاكرة التي يحتاجها المتغير من النوع المصاحب له أو حيز التخزين في

ذاكرة الكمبيوتر مثل Integer عند استخدامه يشغل (٤ bytes)

٢. المدى (Range) هو حدود القيم المتاحة لتخزين المتغير به مثل النوع Byte (يبدأ من صفر وينتهي عند ٢٥٥)

٣. الدقة (Precision) هي عدد الفئات التي يتم تمثيل المتغير بها (وهي خاصة بالمتغيرات الرقمية فقط) وتختلف درجة الدقة باختلاف نوع المتغير

🔑 المتغيرات من النوع (short , integer , long) تستخدم لتخزين الأعداد الصحيحة فقط وليست الأعداد التي تحتوي على كسور

🔑 أما المتغيرات من النوع (single , double , decimal) تستخدم لتخزين الأعداد التي تحتوي على كسور عشرية ولكن بدرجات دقة مختلفة تختلف من نوع إلى آخر

سؤال : ما الفرق بين البيانات الرقمية والبيانات الحرفية ؟

البيانات الرقمية

هي البيانات التي يتم إجراء العمليات الحسابية عليها

البيانات الحرفية

هي البيانات التي لا يجوز إجراء العمليات الحسابية عليها حتى لو كانت هذه البيانات أرقام مثل (رقم الهاتف - رقم المنزل - رقم البطاقة)

٣- إذا أردنا تخزين حرف واحد فقط فإننا نستخدم متغير من النوع char أما إذا أردنا تخزين مجموعة من الحروف نستخدم متغير من النوع string

هام إذا أردنا تخزين الرقم (٢٠٠) فإننا نحتاج إلى بايت واحد (١ byte) أى خلية واحدة لذلك نستخدم متغير من النوع (byte) أما إذا أردنا تخزين القيمة الحرفية " ٢٠٠ " فإننا نحتاج إلى ٣ بايت (٣ byte) لذلك يفضل أن نستخدم متغير من النوع (string)

نذكر أن

- يمكن ضبط الخصائص Properties من نافذة الخصائص Properties window أو من نافذة الكود Code Window
- يمكن إظهار نافذة الخصائص من قائمة View ثم نختار Properties window
- يمكن فتح نافذة الكود من قائمة View الأمر Code

لاحظ

الخصائص Properties هي مكان تخزين البيانات ولها نوع (Type) أو نوع بيان (Data Type) .

- ١- وتختلف الخصائص حسب العنصر النشط فى شاشة الـ IDE
- ٢- يمكن إظهار نافذة الخصائص من خلال لوحة المفاتيح عن طريق الضغط على الزر F4
- ٣- الجزء الأيسر يحتوى على اسم الخاصية Text أما الجزء الأيمن يحتوى على قيمة الخاصية

معرفة نوع البيان (Data type) فى نافذة الكود

١. قف بمؤشر الفأرة على الخاصية Property بعد كتابة جملة التخصيص للخاصية Property فى نافذة الكود
٢. يظهر مستطيل أصفر به نوع البيان Data Type الخاص بهذه الخاصية Property ويظهر بعد كلمة AS

مثال

ضبط خصائص (Text , Auto Size , Width) الخاصة بإداه تحكم العنوان Label1

<pre>Label1.Text = "جمهورية مصر العربية"</pre> <pre>Public Overrides Property Text() As String</pre>	الخاصية (Text) نوعها (String)
<pre>Label1.AutoSize = False</pre> <pre>Public Overrides Property AutoSize() As Boolean</pre>	الخاصية AutoSize نوعها (Boolean)
<pre>Label1.Width = 120</pre> <pre>Public Property Width() As Integer</pre>	الخاصية Width نوعها (Integer)

لاحظ :

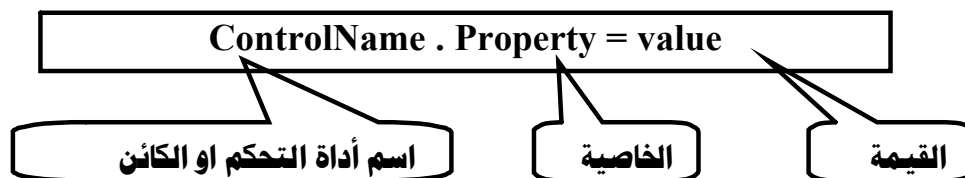
توفر لغة الفيجوال بيزيك أنواع أخرى (Types) غير (Data Types) .

تذكران

القيمة Value التي يتم تخصيصها للخاصية لها عدة أنواع أو أشكال منها:

١. القيمة المجردة مثال الخاصية Text
٢. القيمة المنطقية مثال الخاصية Visible
٣. القيمة من قائمة مثال الخاصية ForeColor
٤. القيمة بإنشاء كائن مثال الخاصية Font
٥. القيمة من ناتج تعبير حسابي
٦. القيمة من متغير Variable أو من خاصية Properties

صيغة ضبط الخصائص properties برمجيا



Label1.BackColor = Color.Red

Public Overridable Property BackColor() As System.Drawing.Color

المتغيرات

تعريفها

هي أماكن مجهزة في ذاكرة الكمبيوتر Ram ولها نوع بيان (Data Types) تتغير أثناء سير تنفيذ أوامر و تعليمات البرنامج

هام

- ١- يتم حجز الأماكن في ذاكرة الكمبيوتر بالإعلان عنها (أي إعطاء أسماء للخلايا التي تحمل قيم متغيرة)
- ٢- يتم تخصيص قيمة للمتغيرات على حسب نوع البيان
- ٣- هذه القيمة يمكن أن تتغير أثناء سير تنفيذ أوامر و تعليمات البرنامج .

لاحظ ان :-

١. قبل استخدام أي متغير داخل برنامج "Visual Basic" لابد من الإعلان عنه
٢. ويتم ذلك باستخدام صيغة معينة تسمى الإعلان عن المتغيرات و هذه الصيغة تبدأ بكلمة Dim

الصيغة العامة للإعلان عن المتغيرات في البرنامج

نستخدم الكلمة Dim ثم كتابة اسم المتغير ثم كلمة AS ثم نوع البيان Data Type كالتالي

نوع المتغير (نوع البيان Data Type) As (variable name) Dim

Dim Population as Integer
 Dim V_Name as String
 Dim Address as String = "Cairo"
 Dim Price as Decimal = 123.5

لاحظ ان :-

1. لابد من استخدام جملة Dim للإعلان عن المتغير
2. يمكن ان تكتب (=) و بعدها قيمة وهذه القيمة ستكون القيمة الابتدائية أو يمكن ان نهملها ولا نستخدم () وفي هذه الحالة

المتغيرات الرقمية تبدأ بقيمة صفر

المتغيرات الحرفية ستكون فارغة

المتغيرات المنطقية تأخذ القيمة False

الإعلان عن مجموعة من المتغيرات باستخدام أمر Dim واحد

Dim ChildWeight , Population ,sum as Integer
 Dim Population as Integer = 123, Name as String = "Ahmed", Price as Decimal = 123.5

قواعد او (شروط) تسمية المتغيرات فى الفيجوال بيرك vb.net

1. يجب أن يبدأ اسم المتغير بحرف من الحروف الإنجليزية (A-Z ، a-z) أو (علامة الشرطة السفلية _ underscore).
2. لا يسمح أن يحتوي الاسم على أي رموز أو علامات خاصة كالأسافة أو النقطة (.) أو علامة الاستفهام (?) أو علامة التعجب (!) أو النجمة أو غيرها (, , + , - , ^ , * , ?) وغيرها
3. يمكن أن يتكون الاسم من حروف وأرقام وعلامة _ underscore الشرطة السفلية .
4. لا يسمح باستخدام الكلمات المحجوزة (reserved words) فى vb.net مثل Byte ، Short ، Data ، double ، Me ، Boolean ، string ، char ، decimal ، single ، Long ، Integer
5. من الأفضل أن يدل الاسم على محتواه. (أختار أسماء ذات معنى مناسب لاستعمال المتغير بحيث يعكس اسم المتغير محتواه و دوره فى الحل مثل المجموع Sum والمتوسط Average <<< أمثلة لأسماء المتغيرة الصحيحة

FirstName
 Ahmed2000
 _Mahmoud
 Spent_mony

أمثلة لأسماء المتغيرة غير الصحيحة

اسم المتغير	سبب عدم الصلاحيه
2ndWorldWar	اسم المتغير يبدأ برقم
Spent.Money	اسم المتغير يحتوى على نقطة
Birth Day	اسم المتغير يحتوى على مسافة
Double	اسم المتغير كلمة من الكلمات المحجوزة فى VB
-Ali	اسم المتغير يبدأ بالشرطة العادية

كيفية اختيار نوع البيان Data Type المناسب لكل متغير

عند اختيار نوع البيان المناسب Data Type يراعى الآتي :

١. حجم البيانات التي سوف تخزن في المتغير Variable أي (حجم الذاكرة "عدد الـ Bytes" الذي يحتاجه البيان الذي سوف يخزن في المتغير)
٢. تحديد نوع بيان القيم Data Type التي سوف يتم تخزينها فيه بناءً على العمليات الحسابية والمنطقية التي من الممكن إجراؤها عليه .

النوع	المقصود به	طبيعة البيانات	السبب
Boolean	منطقي	قيمة true أو false	أحسن استغلال للذاكرة
Object	كائن او العام	كل الأنواع يمكن تخزينها هنا	لأنها أكثر مرونة ونستطيع ان نتكيف
String	حروف	مجموعة من الحروف أو أرقام لا تدخل في الحسابات (أرقام التليفون- الرقم القومي)	الاختيار الوحيد
Short	الأعداد الصحيحة	من -٣٢٧٦٧ الى ٣٢٧٦٧	استهلاك قليل للذاكرة كما انه سريع في الحسابات
Integer		من -٢١٤٧٤٨٣٦٤٨ الى ٢١٤٧٤٨٣٦٤٧+	يحتاج ذاكرة أقل من Long وأسرع في الحسابات
Long		أكبر من ٢١٤٧٤٨٣٦٤٧	الاختيار الوحيد
Single	أرقام تحتوي علي كسور عشرية	من ١٠ ^{-٣٨} الى ١٠ ^{+٣٨} أو سبع خانات دقة	أسرع في العمليات الحسابية من Double
Double		أكبر من ١٠ ^{-٣٨} أو أكثر من سبع خانات دقة	الاختيار الوحيد
Decimal		مبلغ من المال	قدرة علي تخزين الجنيحات والقروش بدقة عالية وبدون أخطاء كما انه سريع في العمليات الحسابية

تدريب 1-1 (الإعلان عن المتغيرات (Variables)

عزيزي الطالب قم بإعداد نافذة النموذج Form الآتية والتي من خلالها نستقبل مدخلات مستخدم

اسم المستخدم جديد

الاسم

تاريخ الميلاد

النوع ☐ أنثى ☐ ذكر

عدد أفراد الأسرة

تسجيل

القيم المتوقع إدخالها من مستخدم البرنامج و نوعها والتسمية المقترحة كما هو موضح بالجدول 1-1

القيم	نوع البيان Data Type	التسمية المقترحة
اسم المستخدم	نص String	U_Name
تاريخ الميلاد	تاريخ Date	U_B_D
النوع	منطقي Boolean	U_Gender
عدد أفراد الأسرة	رقم صحيح Integer	U_C_F

ثم اضغط على زر تسجيل مرتين متتاليتين لفتح نافذة الكود الخاصة بها ثم قم بكتابة معالج الحدث Event Handler والذي يسمى أو يطلق عليه أيضاً إجراء الحدث Event Procedure وقم بالإعلان عن المتغيرات كما بالشكل التالي :

```
Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs) Handles Button1.Click
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer
End Sub
```

لاحظ : معالج الحدث Event Handler يسمى أو يطلق عليه أيضاً إجراء الحدث Event Procedure

لاحظ : أنه بهذه الطريقة يمكن حجز أماكن في ذاكرة الكمبيوتر RAM حتي يمكن لنا تخزين مدخلات مستخدم البرنامج فيها . ويتم ذلك عن طريق ما يسمى بجملة التخصيص . Assignment .

جملة التخصيص Assignment أو الإحلال

١- إن أهم دور للمتغيرات في البرامج هو تخزين البيانات بها ولكي نخزن قيمة في متغير أو خاصية نستخدم امر التخصيص

٢- و هي جملة من طرفين بينهما علامة (=) وهذه العلامة لا يقصد بها علامة تساوي المعروفة في الحساب ولكن يطلق عليها معامل التخصيص

٣- الطرف الأيسر لهذه العلامة هو المكان الذي نوضح فيه القيمة أما الطرف الأيمن عبارة عن القيمة

الصيغة العامة

الطرف الأيمن	علامة التخصيص	الطرف الأيسر
قيمة مجردة أو قيمة من متغير Variable أو خاصية Property أو قيمة من تعبير Expression	=	متغير Variable أو خاصية Property

يجب مراعاة الآتي:-

١. اسم المتغير الذي نريد التخزين فيه يكون على يسار " "
٢. القيمة المراد تخزينها تكون يمين " " وتكون من نوع ملائم لنوع المتغير
٣. ان قيمة المتغيرات التي على يمين " " يتم تخزينها في المتغير الموجود على يسارها
٤. ان قيمة المتغيرات التي على يمين لا تتغير وإنما قيمة المتغيرات الموجودة على شمال هي التي تتغير وايضا ان القيمة الجديدة للمتغير تمل محل القيمة القديمة
٥. القيم المراد تخزينها يجب ان تكون من نوع ملائم للمتغير الذي سيتم التخزين به فالقيم الرقمية يجب ان تخزن في متغيرات رقمية والقيم الحرفية يجب ان تخزن في متغيرات حرفية

أمثلة توضح التخصيص Assignment**قيمة مجردة**

Dim Number As Integer
Number = 5

تم الإعلان عن متغيراً اسمه Number من النوع Integer ثم تم تخصيص القيمة المجردة (5) للمتغير Number

قيمة متغير آخر

Dim Number1 As Integer
Dim Number 2 As Integer
Number1 = 5
Number2 = Number1

تم الإعلان عن متغيرين وهما Number1 و Number2 من النوع Integer ثم تم تخصيص القيمة المجردة (5) للمتغير Number1 و تم تخصيص قيمة المتغير Number1 الى المتغير Number2

قيمة من خاصية

Dim Name As String

Name = Label1.Text

تم الإعلان عن متغير Name من النوع String ثم تم تخصيص القيمة من الخاصية Text لأداة التحكم Label1

قيمة من تعبير حسابي

Dim Number As Single
Number = 7
Number = Number / 3

تم الإعلان عن متغير Number من النوع Single ثم تم تخصيص القيمة المجردة 7 للمتغير Number ثم إعادة عملية التخصيص للمتغير Number من خلال تعبير (Expression) وهو (Number/3)

الإعلان عن المتغير و تخصيص قيمة له أثناء الإعلان في خطوه واحده

Dim Number As Single = 5.6

تم الإعلان عن متغير Number من النوع Single وتم تخصيص القيمة 5.6 أثناء الإعلان عن المتغير Number في خطوة واحدة.

لاحظ:

- يمكن التخصيص للمتغيرات Variables أثناء الإعلان عنها أو في مرحلة لاحقة.
- يمكن إعادة التخصيص أي عدد من المرات ولذلك أطلق لفظ المتغيرات لأن قيمها يمكن أن تتغير أثناء سير عمليات البرنامج في أي وقت.
- التخصيص يمكن أن يكون بقيمة مجردة أو بمتغير أو بخاصية أو بتعبير.
- علامة = لا يقصد بها التساوي الحسابي وإنما يقصد بها علامة للتخصيص كما بالمثال $x = x + 1$
- حيث (x) الموجودة بالطرف الأيسر تمثل المخزن و (x) الموجودة بالطرف الأيمن تمثل القيمة حيث إذا افترضنا أن قيمة (x) تساوي (5) إذا بعد عملية التخصيص يصبح المتغير (x) يساوي (5+1) أي (6)

تخصيص قيم للمتغيرات Variables

سوف نقوم بتخصيص مدخلات مستخدم البرنامج للمتغيرات Variables التي سبق إعلانها في التدريب رقم ١ وهذه المتغيرات هي

U_C_F	U_Gender	U_B_D	U_Name
-------	----------	-------	--------

في هذه الحالة سوف نستخدم خصائص Properties الأدوات Controls التي استقبلت مدخلات مستخدم البرنامج في عملية التخصيص حيث أننا لم نغير قيم خاصية Name لهذه الأدوات كما هو

أداة التحكم	الخاصية
TextBox1	Text
TextBox2	Text
RadioButton1	Checked
TextBox3	Text

جمل التخصيص

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer

    U_Name = TextBox1.Text
    U_B_D = TextBox2.Text
    If RadioButton1.Checked = True Then
        U_Gender = True
    End If
    If RadioButton1.Checked = False Then
        U_Gender = False
    End If
    U_C_F = TextBox3.Text
End Sub
```

لاحظ

تم اختبار الخاصية Checked لأداة التحكم RadioButton1 باستخدام جملة If وسوف يتم شرحها لاحقا ، فإذا كانت الخاصية Checked قيمتها True يخص للمتغير U_Gender القيمة True أما إذا كانت العكس يخص للمتغير U_Gender القيمة False

لاحظ

يوجد اختلاف بين نوع بيان Data Type و الخصائص Properties من المتغيرات المستخدمة Variables كما هو موضح بالجدول التالي:

الخاصية	نوع بيان الخاصية	المتغير	نوع بيان المتغير
TextBox1.Text	String	U_Name	String
TextBox2.Text	String	U_B_D	Date
TextBox3.Text	String	U_C_F	Integer

هام:

- لا يجوز ان يختلف نوع بيان الخصائص من نوع بيان المتغيرات المستخدمة Variables
- لذا تتيح لغة الفيجوال بيزيك امكانية تحويل القيم إلى نوع بيان متوافق مع المتغير Variable أو الخاصية Property المخصص لها القيمة وهو ما يسمى تحويل ضمني

سؤال : ما المقصود بالتحويل الضمني ؟

هو إمكانية توفرها لغة الفيجوال بيزيك لتحويل القيم إلى نوع بيان Data Type متوافق مع المتغير Variable أو الخاصية Property المخصص لها القيمة

استخدام المتغير ا ت Variables

إذا وجد المتغير في الطرف الأيسر من جملة التخصيص فهذا يعني أنه المكان الذي سوف تقرن به القيمة أما إذا وجد في الطرف الأيمن من جملة التخصيص فهذا يعني أنه يعبر عن القيمة الموجودة لديه.

- يمكن استعمال متغير على يمين علامة (=) ومتغير على يسارها في نفس الأمر Price = 50
- حيث المتغير الموجود على يسار العلامة (=) هو مكان تخزين القيمة
- بينما المتغير الموجود على يمينها هو القيمة

Price = Price + 10

تدريب على استخدام المتغير ا ت Variables

الخاصية	القيمة
AutoSize	False
Size, Location	باستخدام مؤشر الفأرة
BorderStyle	FixedSingle

تم بإضافة أداة تحكم عنوان Label

بعد ضبط خصائصها كما هو موضح بالجدول

```
Private Sub Button1_Click(ByVal
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer

    U_Name = TextBox1.Text
    U_B_D = TextBox2.Text
    If RadioButton1.Checked = True Then
        U_Gender = True
    End If
    If RadioButton1.Checked = False Then
        U_Gender = False
    End If
    U_C_F = TextBox3.Text

    Me.Label5.Text = U_Name & vbCrLf & U_B_D & vbCrLf & _
    U_Gender & vbCrLf & U_C_F

End Sub
```

الإعلان عن المتغيرات

تخصيص قيم للمتغيرات

استخدام المتغيرات

قم بفتح إجراء الحدث (Click) الخاص بالزر " تسجيل " ثم أضف الكود (Code) الآتي

```
Me.Label5.Text = U_Name & vbCrLf & U_B_D & vbCrLf & U_gender & _
vbCrLf & U_C_F
```

الاهـظ:

١- الكود السابق عبارة عن أمر تخصيص مجموعة متغيرات كقيمة للخاصية Text لأداة التحكم Label 5 حيث

توجد المتغيرات على يمين علامة (=) إذن فهي القيمة

٢- يفصل بين كل متغير والآخر (& vbCrLf &)

٣- علامة (معامل الربط) (&) تستخدم للربط بين النصوص.

٤- الكلمة المعجزة vbCrLf عبارة عن ثابت حرفي يستخدم لإضافة رمز مفتاح الإدخال وسطر جديد

٥- تستخدم علامة (_) حتى يمكن كتابة سطر الكود على أكثر من سطر في حالة إذا كان سطر الكود Code

طويل بعض الشيء وذلك للتنظيم وتسهيل عملية قراءة الكود Code

٦- كلمة Me من الكلمات المعجزة تعبر عن نافذة النموذج Form الحالية.

بعد الانتهاء من كتابة الكود (Code) نقوم بتشغيل البرنامج أو اختبار البرنامج

١- من قائمة (Debug) اختر (Start Debugging) .

٢- اضغط على أيقونة (Start icon) من شريط الأدوات .

٣- اضغط على (F5) من لوحة المفاتيح (لتشغيله من فيجوال استوديو) .

ونملأ بيانات استمارة التسجيل ثم نقوم بالنقر على مفتاح تسجيل كما هو موضح :

لاحظ :

ظهور محتوى أدوات

التحكم على أداة

العنوان Label 5

كتابة الملاحظات

١- توفر لنا لغة Visual Basic .NET إمكانية كتابة ملاحظات أو تعليقات حيث تساعد القارئ على فهم

وتوضيح الأكواد المكتوبة في نافذة الكود Code Window

٢- يتم كتابة الملاحظات في نافذة الكود Code Window باستخدام الكلمة REM أو العلامة (')

٣- كل ما يكتب بعد ها لا يعتبر أكواد ويهمل عند ترجمة الأكواد Code بواسطة مترجم Compiler اللغة

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Sy
REM الإعلان عن المتغيرات
Dim UserName As String
Dim UserBirthDate As Date
Dim UserGender As Boolean
Dim UserNoFamily As Integer

REM تعيين قيم لهذه المتغيرات
UserName = Me.TextBox1.Text
UserBirthDate = Me.TextBox2.Text
If RadioButton1.Checked = True Then
    UserGender = True
Else
    UserGender = False
End If
UserNoFamily = Me.TextBox3.Text

' استخدام هذه المتغيرات
Me.Label5.Text = UserName & vbCrLf & UserBirthDate & vbCrLf & _
UserGender & vbCrLf & UserNoFamily
End Sub
```

الثوابت Constants

تعريفها

هي عبارة عن أماكن محجوزة بذاكرة الكمبيوتر RAM ولها نوع بيان Data Type ولها قيمة وهذه القيمة لا يمكن أن تتغير أثناء سير تنفيذ أوامر وتعليمات البرنامج

هام

١- يتم الإعلان عن الثوابت ثم تخصيص قيمة لها حسب نوع بيان Data Type أثناء الإعلان عنها فقط

٢- قيمة الثابت Constant (ثابتة) لا يمكن أن تتغير أثناء سير تنفيذ أوامر وتعليمات البرنامج

تعريف آخر

الثوابت : هي مخزن في ذاكرة الكمبيوتر يحمل القيم التي لا تتغير أثناء تشغيل البرنامج ويجب الإعلان عنها داخل البرنامج وإعطاء قيمة لها أثناء الإعلان عنها

الصيغة العامة للإعلان عن الثوابت في البرنامج

نستخدم جملة Const ثم كتابة اسم للثابت Constant ثم كلمة As ثم نوع البيان Data Type ثم علامة = ثم القيمة كما هو مبين كالآتي

القيمة Value = نوع الثابت (نوع البيان Data Type) As (constant name) اسم الثابت Const

أمثلة

١- الإعلان عن ثابت C_Name له نوع بيان String وتم تخصيص القيمة النصية "جمهورية مصر العربية" أثناء الإعلان

Const C_Name As String = " جمهورية مصر العربية "

٢- الإعلان عن ثابت pi له نوع بيان Single وتم تخصيص القيمة الرقمية 22 / 7 أثناء الإعلان.

Const pi As Single = 22 / 7

٣- الإعلان عن ثابت BirthDate له نوع بيان Date وتم تخصيص قيمة التاريخ #1/25/2011 أثناء الإعلان.

Const BirthDate As Date = #1/25/2011#

لاحظ

نستخدم علامتي " " في حالة إذا ما أردنا كتابة نص
نستخدم علامتي # # في حالة إذا ما أردنا كتابة تاريخ أو وقت

تذكر فتح نافذة الكود

١- اضغط على مفتاح F7 من لوحة المفاتيح

٢- قم بتنشيط الأداة المراد فتح نافذة الكود لها ثم نختار من قائمة View الأمر Code

٣- قم بالضغط مرتين متتاليتين D- Click على الأداة المراد فتح نافذة الكود لها

٤- اضغط بزر الفأرة الأيمن ثم اختر من القائمة المختصرة View Code

تدريب الإعلان عن الثوابت Constants

١- سوف نقوم بإعداد نافذة نموذج

Form كما هو موضح

٢- ثم نقوم بفتح نافذة الكود Code window بالضغط على مفتاح F7 ثم

إضافة إجراء الحدث Click لكل من زري الأمر Button1, Button2

```
Public Class Form3
    Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs)
        End Sub
    Private Sub Button2_Click(ByVal sender As Object, ByVal e As EventArgs)
        End Sub
End Class
```

علماً بأن :

مساحة الدائرة πR^2 محيط الدائرة $2\pi R$

حيث R تمثل نصف القطر

 $\pi \approx 22/7$

لاحظ :

سوف نكتب الكود Code الخاص بحساب مساحة الدائرة باستخدام إجراء الحدث Button1_Click ومحيط الدائرة باستخدام إجراء الحدث Button2_Click ونظهر النتيجة داخل أداة التحكم عنوان Label2 عند النقر على أحدهما.

في نطاق إجراء الحدث Button1 Click نكتب الكود Code التالي :

```
Private Sub Button1_Click(ByVal sender As Object, ByVal e As EventArgs)
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius ^ 2
End Sub
```

شرح الكود :

١. ثم الإعلان عن متغير Radius من النوع Single ←

٢. ثم الإعلان عن ثابت pi من النوع Single وتم وضع القيمة 22/7 ←

٣. ثم تعيين القيمة من الخاصية TextBox1.Text للمتغير Radius ←

٤. ثم تعيين التعبير $\pi * \text{Radius}^2$ للخاصية Label2.Text ←

تطبيقاً للقانون الرياضي الخاص بمساحة الدائرة

بعد الانتهاء من كتابة الكود Code نقوم بالضغط

على مفتاح F5 لعمل Start Debugging

ونكتب قيمة نصف القطر ثم نقوم بالنقر على زر

"مساحة الدائرة"

تدريب (5) تحديد نطاق إعلان المتغيرات (Variables) والثوابت (Constants)

يقصد بنطاق المتغيرات أو الثوابت موضع الإجراء الذي يتم فيه الإعلان عن الثابت أو المتغير واستخدامه.

قم بإضافة الكود Code الخاص بحساب محيط الدائرة وذلك في نطاق إجراء الحدث Button2_Click فكتب الكود Code الآتي:

Radius = TextBox1.Text

Label2.Text = pi * Radius * 2

Private Sub Button1_Click(ByVal sender As Object) ليصبح شكل الكود كما يلي.

Dim Radius As Single

Const pi As Single = 22 / 7

Radius = TextBox1.Text

Label2.Text = pi * Radius ^ 2

End Sub

Private Sub Button2_Click(ByVal sender As Object)

Radius = TextBox1.Text

Label2.Text = pi * Radius * 2

End Sub

كما يظهر المستطيل الأصفر عند توجيه مؤشر الفأرة إلى المتغير pi في السطر الثاني كما هو موضح

Private Sub Button2_Click(ByVal sender As Object)

Radius = TextBox1.Text

Name 'pi' is not declared.

Label2.Text = pi * Radius * 2

End Sub

لا حظ :

- ١- أننا لم نعلن عن المتغير والثابت في نطاق إجراء الزر Button 2
- ٢- تم الاعتماد على أن المتغير Radius والثابت pi سبق إعلانهما في نطاق إجراء الحدث Button1_Click
- ٣- تظهر رسائل داخل مستطيل أصفر تنبه أن المتغير Radius والثابت pi لم يتم الإعلان عنهما رغم أنهما معلنين في نطاق إجراء الحدث Button1_Click
- ٤- أن المتغيرات والثوابت لا نستطيع استخدامها إلا في نطاق إعلانها

ولحل هذه المشكلة

يمكن أن نعيد إعلان المتغير Radius والثابت pi مرة أخرى في نطاق إجراء حدث الزر Button2_Click لأنه لا يمكن استخدامهما إلا في حدود نطاق الإجراء فقط

```
Private Sub Button1_Click(ByVal sender As Object,
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius ^ 2
End Sub
```

```
Private Sub Button2_Click(ByVal sender As Object,
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius * 2
End Sub
```

اول حل هذه المشكلة

كما يمكننا أن نعلن عن المتغيرات Variable والثوابت Constant على مستوى التصنيف Class ومندها لا نحتاج عملية الإعلان عند كل نطاق إجراء حدث Event Procedure كما هو موضح

```
Public Class Form3
```

```
    Dim Radius As Single
    Const pi As Single = 22 / 7
```

```
    Private Sub Button1_Click(ByVal sender As
        Radius = TextBox1.Text
        Label2.Text = pi * Radius ^ 2
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As
        Radius = TextBox1.Text
        Label2.Text = pi * Radius * 2
    End Sub
```

```
End Class
```

الشرح

حيث تم الإعلان عن المتغير Radius والثابت pi في نطاق التصنيف Form3 فاستطعنا استخدامهما داخل نطاق إجراء أحداث Button1_Click و Button2_Click لأن إجراء أحداث Button1_Click و Button2_Click معلنه في نطاق التصنيف Form3

نطاق المتغيرات :

يحدد نطاق " مجال " المتغير أى الأماكن المسموح بها استعمال المتغير داخل البرنامج وتنقسم المتغيرات من حيث النطاقات أو المجالات الى نوعان

١- المتغيرات المحلية (Local Variables) :

وهي متغيرات يتم الإعلان عنه داخل إجراء حدث معين مثل الحدث Button1_Click وينفذ داخل هذا الإجراء فقط ويفقد قيمته إذا ما تم استعماله خارج هذا الإجراء مثل الحدث Button2_Click

٢- المتغيرات العامة " الشاملة " (Global Variables) :

هي متغيرات يتم الإعلان عنه خارج جميع الإجراءات نطاق التصنيف Form3 وهي يمكن استخدامها في كل إجراءات البرنامج ولا يفقد قيمتها عند انتهاء الإجراء المستخدمة به

الأخطاء Errors

لابد من الإلتزام بقواعد لغة البرمجة المستخدمة حيث أنه عند كتابة الكود Code يحدث عدة أنواع من الأخطاء منها

أنواع الأخطاء		
الإملائية والنحوية Syntax Errors	المنطقية Logic Error	عند التشغيل Runtime Error

١- الأخطاء والنحوية والإملائية Syntax Errors

هي الأخطاء التي تحدث عند كتابة الكود Code بصورة غير سليمة كما هو موضح بالأمثلة الآتية:

أمثلة

Dim x As Single

تم الإعلان عن المتغير x ولكن كتبت كلمة Dim خطأ

Const x As Single

تم الإعلان عن الثابت x ولكن لم تخصص له قيمة أثناء الإعلان

كيفية التغلب على الأخطاء الإملائية والنحوية

للتغلب على هذا النوع من الأخطاء يساعدنا IDE بيئة التطوير المتكاملة بأنها لا تسمح بوجود أي خطأ من هذا النوع. وتظهر خطأ متعرج تحت الخطأ

٢- الأخطاء المنطقية Logic Error

هي الأخطاء التي تؤدي إلى نتائج غير سليمة عند استخدام البرنامج وتحدث إذا استخدمت تعبيرات Expressions بناؤها غير سليم في جمل التخصيص غالباً

عند حساب مساحة الدائرة استخدمنا الكود الآتي:

أمثلة

Dim Radius As Single

Const x As Single = 22 / 7

Radius = TextBox1.Text

Label2.Text = x * Radius ^ 2

Dim Radius As Single

Const x As Single = 22 / 7

Radius = TextBox1.Text

Label2.Text = x + Radius ^ 2



عند تنفيذ الكود Code سوف يعمل دون أن يعطي أي رسائل خطأ ولكن النتيجة سوف تصبح مغايرة للواقع لأنه سوف يتم الجمع بدلا من الضرب.

كيفية التغلب على الأخطاء المنطقية Logic Error

١- يجب مراعاة الكود المكتوب جيدا

٢- اختبار البرنامج على بيانات سبق التأكد من صحتها حتى يتم التأكد من عدم وجود أخطاء من النوع Logic Error

٢- الأخطاء عند التشغيل Runtime Error

هي الأخطاء التي تظهر أثناء التشغيل ونجدها غالبا في الأكواد Code التي يتم فيها جملة تخصيص Assignment

متى تحدث :

١. تحدث عند تخصيص قيمة أكبر من مدى نوع البيان المستخدم

٢. تحدث عند تخصيص قيمة نوعها يختلف عن نوع بيان المتغير أو الثابت أو الفاصية ولا يستطيع مترجم Compiler اللغة تحويلها لهذا النوع Type

أمثلة على الأخطاء التي تظهر أثناء التشغيل

٢- أكتب الكود Code الآتي لدى إجراء

الحدث Click للزر "مجموع"

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    'الإعلان عن متغير لكل مادة'
    Dim arabic As Byte
    Dim computer As Byte

    'تخصيص مدخلات المستخدم لكل متغير'
    arabic = TextBox1.Text
    computer = TextBox2.Text

    'تخصيص ناتج مجموع المتغيرين'
    Label3.Text = arabic + computer
End Sub
```

١- قم بإعداد نافذة النموذج كما هو موضح بشكل



لاحظ:

تم الإعلان عن متغيرين arabic (و) computer لهما نوع البيان byte والقيم التي يمكن أن يستقبلها تبدأ من صفر وتنتهي عند 255 وهذا يعني أنه يجب على مستخدم البرنامج أن يلتزم بحدود تلك القيم وفي حالة مخالفتها يحدث خطأ أثناء التشغيل Runtime عند إجراء التخصيص لهذه المتغيرات Variables كما يظهر في الحالات الآتية:

الحالة الأولى:

١- قم بعمل Start Debugging وذلك بالضغط على مفتاح F5

ثم قم بالنقر على مفتاح "مجموع" دون إدخال أي بيانات

لصناديق النص TextBox



٢- تظهر لنا رسالة الخطأ الآتية أنه لا يجوز تحويل القيمة النصية الفارغة "" إلى النوع Byte وحدث ذلك بسبب أننا لم ندخل درجة اللغة العربية

رسالة خطأ أثناء وقت التشغيل بسبب عدم ادخال أي قيمة

تخصيص مدخلات المستخدم لكل متغير

```
arabic = TextBox1.Text
computer = TextBox2.Text
```

يتم ناتج مجموع المتغيرين

```
Label3.Text = arabic + computer
```

InvalidCastException was unhandled

Conversion from string "" to type 'Byte' is not valid.

الحالة الثانية

قم بإيقاف التشغيل Stop Debugging وذلك بالضغط على أيقونة

وذلك بالضغط على مفتاح F5 ثم قم بالنقر على مفتاح مجموع بعد إدخال قيمة نصية (خمس عشرة درجة) لصندوق

النص TextBox

كمبيوتر خمسة عشر

نظهر لنا رسالة الخطأ الآتية أنه لا يجوز تحويل القيمة النصية "خمس عشرة درجة" إلى النوع Byte حدث ذلك بسبب إدخال

قيمة نصية بدلا من قيمة رقمية

رسالة خطأ أثناء وقت التشغيل بسبب إدخال نص.

تخصيص مدخلات المستخدم لكل متغير

```
arabic = TextBox1.Text
computer = TextBox2.Text
```

يتم ناتج مجموع المتغيرين

```
Label3.Text = arabic + computer
```

InvalidCastException was unhandled

Conversion from string "خمس عشرة درجة" to type 'Byte' is not valid.

الحالة الثالثة

قم بإيقاف التشغيل Stop Debugging وذلك بالضغط على أيقونة

وذلك بالضغط على مفتاح F5 ثم قم بالنقر على مفتاح مجموع بعد إدخال قيمة أكبر من 255 لصندوق النص TextBox

رسالة الخطأ الآتية نتيجة إدخال قيمة أكبر من الحد المسموح وحدث ذلك بسبب إدخال درجة الكمبيوتر أكبر من 255 أي

القيم خارج الحدود المسموحة

تخصيص مدخلات المستخدم لكل متغير

```
arabic = TextBox1.Text
computer = TextBox2.Text
```

تخصيص ناتج مجموع المتغيرين

```
Label3.Text = arabic + computer
```

كمبيوتر 300

OverflowException was unhandled

Arithmetic operation resulted in an overflow.

لاحظ:

الشكل البيضاوي المرسوم على كل رسالة خطأ يشير إلى نوع الخطأ الذي لم يتم معالجته.

كيفية التغلب على أخطاء التشغيل Runtime Error

الطريقة الأولى

توفر لغة Visual Basic .NET إمكانية معالجة الأخطاء المحتمل وقوعها من خلال جملة Try/ Catch

مجموعة أوامر Try / Catch

نستخدم لاكتشاف الأخطاء التي تحدث أثناء تشغيل البرنامج والتعامل معها

١. يقوم VB . net باكتشاف الأخطاء و يقوم بعرض رسائل تفيد بحدوث خطأ

٢. لكن المبرمجين يفضلون استعمال Try / Catch

□ لأنها تتيح لهم التعامل مع الأخطاء بأنفسهم

□ عرض رسائل أوضح لمستخدم البرنامج يقوم المبرمج بكتابتها

الصيغة العامة لها :-

Try

الكود Code المحتمل أن يظهر منه خطأ أثناء التشغيل

Catch ex As Exception

كود Code الخاص بمعالجة الخطأ إذا وقع وغالباً يكون رسالة

EndTry

بقية أوامر البرنامج

شرح الأمر

١. إذا قمنا بتنفيذ الأوامر بعد Try بدون إخطاء ينتقل البرنامج إلى الأوامر بعد End Try مباشرة ويتم تجاهل

الأوامر التي تلي Catch

٢. أما إذا تم تنفيذ الأوامر بعد Try بوجود إخطاء ينتقل البرنامج إلى الأوامر بعد Catch ليعرض الرسالة الموجودة

بها ولا يستكمل تنفيذ ما بقي من الأوامر التي تلي Try

الإعلان عن متغير لكل مادة

Dim arabic As Byte

Dim computer As Byte

Try

تخصيص مدخلات المستخدم لكل متغير

arabic = TextBox1.Text

computer = TextBox2.Text

Catch ex As Exception

Label3.Text = "من فضلك أدخل الدرجات بصورة صحيحة"

Exit Sub

End Try

تخصيص ناتج لجمع المتغيرين

Label3.Text = arabic + computer

الطريقة الثانية

التحكم في مدخلات المستخدم بحيث يوفر للمستخدم الأدوات Controls مع ضبط خصائصها Properties التي من خلالها يقوم بإدخال البيانات الصحيحة دون غيرها
(أى إلزام المستخدم على الاختيار من ضمن البيانات الصحيحة المعروضة فقط دون كتابة أى بيانات أخرى)

لاحظ :

في نافذة النموذج السابقة لا يستطيع المستخدم اختيار دولة ليست موجودة في القائمة.

لاحظ :

وهذه أدوات تقدمها لغة البرمجة للمبرمج ليفرض بها اختيارات محددة للمستخدم حتى يضمن تقليل عدد الأخطاء إلى أدنى حد

الجدول الآتي يوضح العمليات الحسابية التي يمكن إجراؤها في vb.net

العمليات الحسابية

العملية	جمع	طرح	ضرب	قسمة عادية	باقي القسمة	أسس
المعامل المؤثر	+	-	*	/	mod	^

أولويات تنفيذ العمليات الحسابية

جميع لغات البرمجة بما فيها vb.net قد وضعت قاعدة تسمى قاعدة أولويات تنفيذ العمليات الحسابية لتحديد أولوية تنفيذ العملية الحسابية وفي هذه القاعدة يتم ترتيب المعاملات المختلفة ترتيباً تنازلياً حسب أولوية تنفيذها عند اجتماعها في عملية حسابية واحدة

فمثلاً إذا أجرى بعض عمليات التخصيص ومنها تخصيص تعبير حسابي مثل

Label2.Text = x * Radius * 2

في حالة إذا ما وجد أكثر من عملية حسابية مثل

5+3*2

أيهما ينفذ أولاً عملية الضرب أم عملية الجمع ؟

يتم تنفيذ عملية الضرب أولاً ثم يجمع الناتج بحيث تصبح النتيجة (11) ولكن إذا وضعنا أقواس بهذه الطريقة (5+3) * 2 ينفذ ما بداخل القوس أولاً فتصبح النتيجة (16)

تنفذ العمليات الحسابية وفقا للترتيب الآتي

١. العمليات التي بين الأقواس (الداخلية أولا ثم الخارجية)
٢. عمليات الأسس
٣. عمليات الضرب و القسمة (الأولوية من اليسار ثم اليمين)
٤. عمليات الجمع و الطرح (الأولوية من اليسار إلي اليمين)

ملحوظة هامة جدا

في حالة وجود أكثر من عملية لها نفس الأولوية كالضرب والقسمة او الجمع والطرح فان ترتيب تنفيذها يكون من اليسار الى اليمين ، أما إذا تداخلت الأقواس (كأن يكون هناك أقواس داخل أقواس) فان الأقواس الداخلية يتم تنفيذ ما بينها أولا

خلى بالك : في حالة وجود عمليات حساب باقي القسمة فانها تأتي بعد الضرب و القسمة و قبل الجمع و الطرح

ملحوظة هامة جدا

من الضروري مراعاة أولويات تنفيذ العمليات الحسابية عند إنشاء تعبيرات Expression تستخدم في جملة التخصيص Assignment حتى لا تقع أخطاء من النوع Logic Errors لأن النتيجة ستتغير في هذه الحالة دون ظهور أي رسائل خطأ

مثال :-

ما هي نتيجة تنفيذ العمليات الآتية بافتراض القيم التالية للمتغيرات :-

$$X = 2 \quad Y = 3 \quad Z = 4$$

- A) $X * (Y + 1)$
- B) $Y \wedge X * X + Z * 3$

(A) الحل

١- نستبدل كل متغير بقيمته

$$2 * (3 + 1)$$

٢- حساب ما بداخل الأقواس أولا

$$2 * (4)$$

٣- تنفيذ عملية الضرب فيكون الناتج (8)

(B

١. نستبدل كل متغير بقيمته $3 \wedge 2 * 2 + 4 * 3$

٢. يتم حساب الاس أولا $9 * 2 + 4 * 3$

٣. تنفيذ عملية الضرب التي في اليسار $18 + 4 * 3$

٤. تنفيذ عملية الضرب التي في اليمين $18 + 12$

٥. تنفيذ الجمع ليكون الناتج 30

أسئلة وتدريبات

السؤال الأول:

في نافذة النموذج الآتية:

تسجيل مستخدم جديد

الاسم: مرودة ماهر

تاريخ الميلاد: 20 يونيو , 1998

النوع: ☒ انثى ☐ ذكر

الجنسية: مصر

تسجيل

إذا أردنا أن نخزن مدخلات المستخدم في متغيرات حدد نوع البيان المناسب لكل مدخل من مدخلات مستخدم البرنامج:

- ١
- ٢
- ٣
- ٤

السؤال الثاني:

في الكود (Code) الآتي:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Radius As Single
    Const x As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = x * Radius ^ 2
End Sub
```

استخرج أسماء المتغيرات والثوابت وأنواعها:

السؤال الثالث:

في الأكواد الآتية حدد سبب الخطأ

تخصيص مدخلات المستخدم لكل متغير

```
arabic = TextBox1.Text
computer = TextBox2.Text
```

يتم ناتج مجموع المتغيرين

Label3.Text = arabic + computer

InvalidCastException was unhandled

Conversion from string "خمس عشرة درجة" to type 'Byte' is not valid.

السبب:

تخصيص مدخلات المستخدم لكل متغير

```
arabic = TextBox1.Text
computer = TextBox2.Text
```

تخصيص ناتج مجموع المتغيرين

Label3.Text = arabic + computer

Sub

OverflowException was unhandled

Arithmetic operation resulted in an overflow.

السبب:

السؤال الرابع:

حدد ماهو الكود (Code) المطلوب كتابته حسب ما تشير اليه الأسهم:

Try



Catch ex As Exception



EndTry

السؤال الخامس:

حدد نوع الخطأ في الكود (Code) الآتي ثم صوبه

الكود (Code)	نوع الخطأ	التصويب
Din x As Single		
Const x As Single		

السؤال السادس:

عند كتابة الكود يحتاج المبرمج لوضع تعليق معين لا يتم تنفيذه، لابد أن يسبقه :

- ٢

أو

- ١

السؤال السابع:

في الكود (Code) الآتي:

```
Private Sub Button1 Click(ByVal sender As System.Object, ByVal e As S
Dim UserName As String
Dim UserBirthDate As Date
Dim UserGender As Boolean
Dim UserTelNo As Integer

UserName = TextBox1.Text
UserBirthDate = TextBox2.Text
UserGender = CheckBox1.CheckState
UserTelNo = TextBox3.Text
```

١

٢

فسر الكود المشار إليه بالأرقام على الشكل

-١

-٢

السؤال الثامن:

رتب السطور الآتية وفقا لأولويات تنفيذ العمليات الحسابية

العملية	الترتبة
الضرب والقسمة من اليسار إلى اليمين.	()
الأقواس من الداخل إلى الخارج.	()
الجمع والطرح من اليسار إلى اليمين.	()
الأسس.	()

السؤال التاسع:

حدد ناتج تنفيذ العمليات الحسابية الآتية:

أ - $(5+3)*2$

النتيجة:

ب - $5+3*2$

النتيجة:

السؤال العاشر:

ضع علامة (✓) أمام العبارة الصحيحة، وعلامة (✗) أمام العبارة الخطأ:

(١)	أحد قواعد تسمية المتغيرات أو الثوابت في البرنامج أن يبدأ اسم المتغير بحرف أو رقم. (...)
(٢)	يتم الإعلان عن المتغيرات باستخدام جملة Dim. (...)
(٣)	المتغير من النوع Double يأخذ القيمة True أو False. (...)
(٤)	المتغيرات من أنواع (Integer & Long & Double) تستخدم لتخزين الأعداد الصحيحة فقط. (...)
(٥)	يتم استقبال مدخلات المستخدم من خلال العديد من الأدوات منها TextBox. (.....)