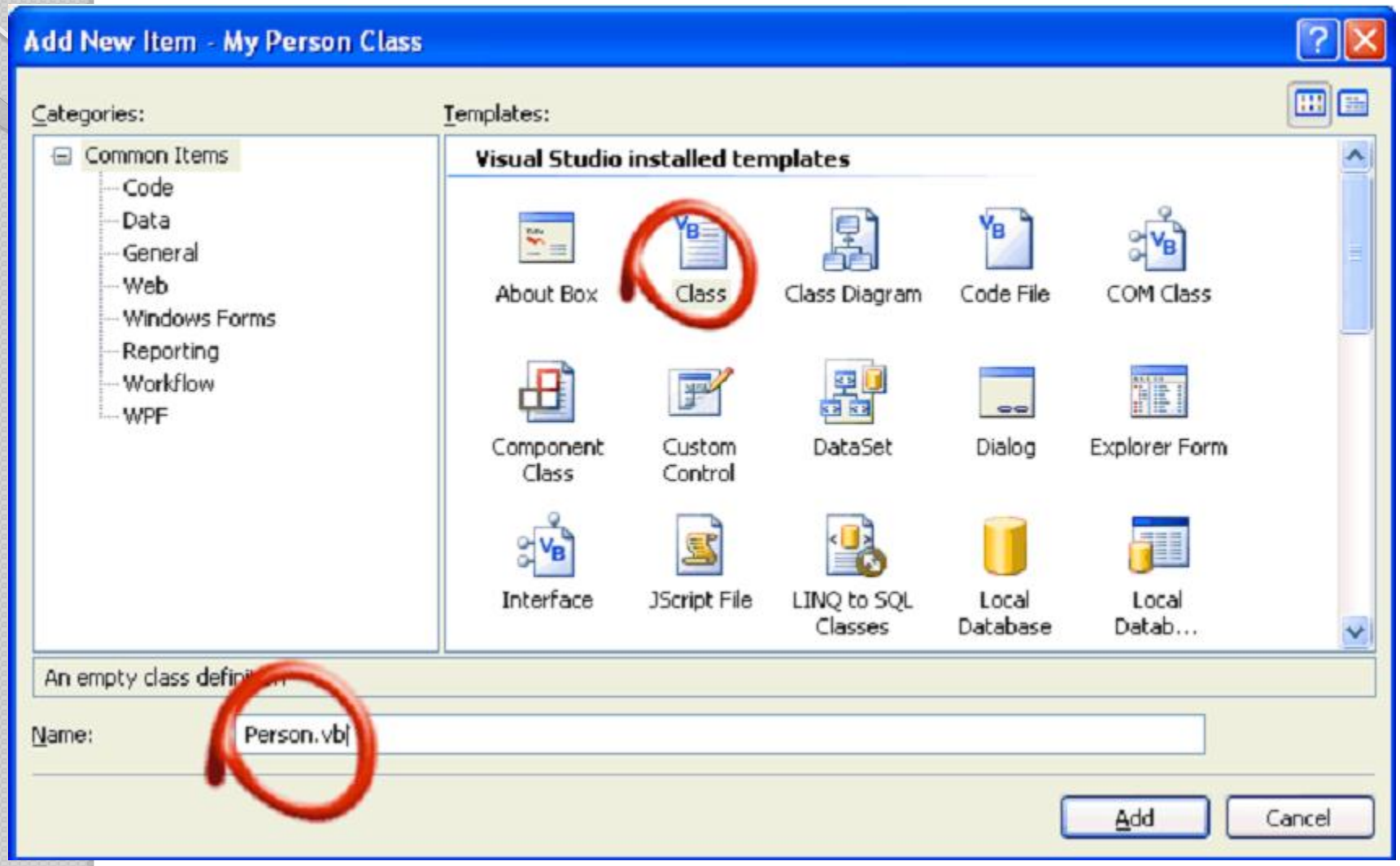
المحاضرة الثالثة

نقوم بإضافة فئة Class للمشروع، نختار Add ثم New Item ثم نختار من النافذة التي تظهر الفئة Class ونسميها Person.vb ثم نضغط Add.



سيقوم الفيجوال بيسك بإضافة الفئة إلى ملفات المشروع ثم سيقوم بفتح صفحة الكود الخاصة بالفئة، وسنقوم الآن بكتابة الأكواد الخاصة بالفئة Person بحسب الترتيب التالي: تعريف متغيرات الفئة ثم كتابة الخصائص ثم الطرق Methods ثم إنشاء الكائنات التى تعتمد على هذه الفئة.

دعونا لنبدأ بكتابة كود الفئة خطوة بخطوة:

١- تعريف متغيرات الفئة

لابد أن نقوم بكتابة جميع الأكواد بين Public Class Person و End Class ولتعريف المتغيرات لابد أن نقوم بتعريفهم بعد Public Class Person مباشرة، فنكتب بعدها في السطر الثاني مباشرة التعريفات:

```
Private Name1 As String
Private Name2 As String
```

لقد قمنا بتعريف المتغيرين بالكلمة Private لكي نستخدمهم بداخل الفئة Person فقط وغير قابلين للاستعمال على طول المشروع

٢- إنشاء الخصائص:

بعد تعريف المتغيرات لابد من إنشاء الخصائص، نكتب هذا الكود بعد تعريف المتغيرات:

```
Public Property FirstName() As String
```

بمجرد أن نضغط Enter بعد كتابة الجملة أعلاه يقوم الفيجوال بيسك بكتابة كود تلقائي من أجل تعبئته، فنحن قمنا بكتابة خاصية فقام الفيجوال بيسك بكتابة بقية كود الخاصية الذي لابد علينا من تعبئته لنقوم بتعريف الخاصية بشكل صحيح، بقية الكود يحتوي على بند Get والتي تعني ماذا سوف يرى المبرمج عند استخدام الخاصية FirstName، وكذلك بند Set والذي يُحدد ماذا يَحدث إذا قام المبرمج بتغيير قيمة الخاصية FirstName وفي نهاية كود الخاصية توجد الجملة End Property والتي توضح انتهاء كود الخاصية.

٣- نقوم الآن بتعبئة بقية الكود التابع للخاصية FirstName كالتالي:

```
Public Property FirstName() As String
    Get
        Return Name1
    End Get
    Set(ByVal value As String)
        Name1 = value
    End Set
End Property
```

مع ملاحظة بأننا نقوم بكتابة الكود المظلل بالأصفر فقط. وكما وضحنا من قبل فـــ Get تعني ماذا سوف يشاهد المبرمج عند اختياره للخاصية، و Set تعني ماذا يحدث عند تغيير الخاصية، وبنفس الطريقة نقوم بكتابة خاصية ثانية LastName كالتالي فبعد End Property نضغط Enter لنكتب الخاصية الجديدة كالتالي: (نفس الخاصية السابقة عدا التغييرات المظللة بالأصفر)

```
Public Property LastName() As String
    Get
        Return Name2
    End Get
    Set(ByVal value As String)
        Name2 = value
    End Set
End Property
```

٤- إنشاء الطُرق Methods

تحت الخاصية LastName نكتب تعريف الدالة Age كالتالي:

```
Public Function Age(ByVal Birthday As Date) As Integer
Return Int(Now.Subtract(Birthday).Days / 365.25)
End Function
```

تقوم الدالة بحساب عدد الأيام من تاريخ الميلاد إلى اليوم ثم قسمة المجموع على ٣٦٥,٢٥ (لإن عدد أيام السنة يساوي ٣٦٥ والسنوات الكبيسة (رابع سنة) تساوي ٣٦٦ لذلك قام المؤلف بالقسمة على ٣٦٥,٢٥ واستخدم Int لتقريب الكسر لأقرب رقم). لاحظ الدالة بين القوسين (Now.Subtract(Birthday حيث يقوم البرنامج بطرح تاريخ الميلاد من التاريخ الحالي ثم تتم القسمة فيما يعد على ٣٦٥,٢٥ ثم بعدها تتم عملية التقريب.

```
Public Class Person
    Private Name1 As String
    Private Name2 As String

    Public Property FirstName() As String
        Get
            Return Name1
        End Get
        Set(ByVal value As String)
            Name1 = value
        End Set
    End Property

    Public Property LastName() As String
        Get
            Return Name2
        End Get
        Set(ByVal value As String)
            Name2 = value
        End Set
    End Property


    Public Function Age(ByVal Birthday As Date) As Integer
        Return Int(Now.Subtract(Birthday).Days / 365.25)
    End Function

End Class
```

ننقر نقرتين على الزر Button1 ونكتب الكود التالي في الحدث Click التابع للزر:

```
Dim Employee As New Person
Dim DOB As Date
Employee.FirstName = TextBox1.Text
Employee.LastName = TextBox2.Text
DOB = DateTimePicker1.Value.Date
MsgBox(Employee.FirstName & " " & Employee.LastName _
& " عمره " & Employee.Age(DOB) & "سنة ")
```

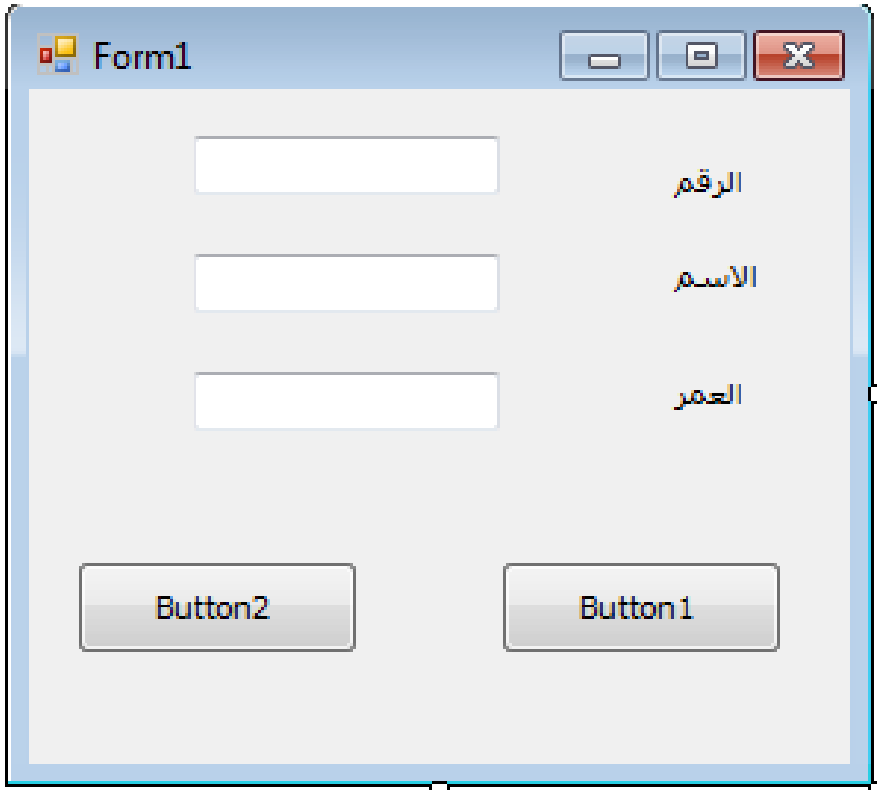
بإسناد النص في مربع النص TextBox1 إلى المتغير FirstName ثم قمنا بإسناد النص في مربع النص الثاني TextBox2 إلى المتغير LastName، ثم قمنا بسحب التاريخ في DateTimePicker1 إلى المتغير DOB ، بعدها قمنا بكتابة كود لإظهار صندوق حوار يحتوي على الاسم الأول والأخير ثم العمر بالاعتماد على الفئة Person وبالأخص على الدالة Age الموجودة في الفئة المذكورة، حيث قمنا بإضافة التاريخ الذي قام باختياره المستخدم إلى المتغير DOB ثم وضعناه بدل المتغير BirthDay الموجود في الدالة Age الموجودة في الفئة

الفئة Person
قم بإضافة الأسم الأول والأخير وتاريخ الميلاد
محمد
حمدون
16 اغسطس 1990
عرض السجل
My Person Class
محمد حمدون عمره ١٩ سنة
OK

مثال:

انشئ الـ form الموضح مع انشاء

- class person1 يحتوي علي المتغير الخاص name ، المتغير العام age .
- Class student يحتوي علي المتغيرات العامة name , id ، الدالة hello لطباعة name.
- الزر button1 لتعريف كائن من نوع person1 و ملئ بياناته و طباعتها.
- الزر button2 لتعريف كائن من نوع studentو ملئ بياناته و طباعتها.



```
Public Class person1
    Private name As String
    Public age As Integer
    Public Property personname() As String
        Get
            Return name
        End Get
        Set(ByVal value As String)
            name = value
        End Set
    End Property
End Class
Public Class student
    Public id As Integer
    Public name As String
    Public Sub hello()
        MsgBox("hi, " & name)
    End Sub
End Class
```

کود form1

```
Public Class Form1
    Dim p As person1
    Dim s As student

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Sys
        p = New person1
        p.age = TextBox3.Text
        p.personname = TextBox2.Text
        MsgBox(p.personname & "   " & p.age)

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As Sys
        s = New student
        s.id = TextBox1.Text
        s.name = TextBox2.Text
        MsgBox(s.name & "   " & s.id)
    End Sub
End Class
```

دوال البناء (المشيدات) Constructors

- تعمل تلقائيا عند انشاء (اشتقاق) كائن جديد من الفصيلة.
- تقوم ببناء كائن فارغ أو ممتلئ بالبيانات (قيم اعضاء الطبقة).
- تأخذ الاسم new.

```
Sub new()
    ..............
End sub
```

مثال:

```
Public Class Class1
    ......
    Sub new()
        MsgBox("object is built")
    End sub
    ......
End class
```

اشتقاق كائن جديد

```
Dim X as New Class1
```

يتم استدعاء الاجراء new لينفذ ما بداخله.

- ممكن انشاء دالة بناء new لبناء كائن فارغ أو انشاء دالة بناء لإنشاء كائن مع ملئه بالبيانات (دالتي بناء باستخدام التحميل الزائد للدوال).

المشيدات Constructors

عندما نقوم بتعريف new Person() فإن هذا يعني اننا نقوم بتشغيل الدالة ال Constructor، وهي الدالة التي تعمل مع تشغيل اي نسخة من البرنامج ، افتراضياً تكون هذه الدالة خالية ويمكننا وضع بعض الاوامر فيها التي نحتاجها وقت انشاء نسخة ، لعرض مثلاً رسالة تخبرنا بانشاء نسخة جديدة من الفئة Class :

```
Public Sub New()
    Console.WriteLine("new object")
End Sub
```

يمكن ايضاً ان يستقبل ال Constructor بارميترس، فمثلاً لجعل الConstructor يستقبل الاسم الأول مع تعريف الاوبجكت الجديد ، فسيكون ذلك بالشكل التالي:

```
Public Sub New(ByVal userfirstname As String)
    FirstName = userfirstname
End Sub
```

ولعمل نسخة جديدة سيتوجب علينا كتابة الكود التالي:

```
Dim MyEmpolyee As New Person("ahmed")
```

انشاء أكثر من دالة بنفس الاسم

١. باستخدام المعاملات الافتراضية.
٢. التداخل overloading.

التداخل overloading

تعريف اكثر من دالة بنفس الاسم داخل نفس الطبقة مع اختلاف في عدد/نوع/ترتيب المعاملات.

مثال:

- انشاء فصيلة (طبقة) shape تحتوى على ٣ دوال drawline.
- الدالة الاولى لا يمرر اليها معاملات و تقوم بطباعة الحرف * عدد ٢٠ مرة.
- الدالة الثانية يمرر اليها معامل واحد (عدد مرات طباعة الحرف *).
- الدالة الثالثة يمرر اليها معاملان (عدد مرات طباعة الحرف ، الحرف المراد طباعته).

```
Public Class shape
   public overloads sub drawline()
      dim I as integer
      console.writeline("first method")
      for i=0 to 20
         console.write("*")
      next
      console.writeline()
   end sub
   public overloads sub drawline(byval n as integer)
      dim I as integer
      console.writeline("second method")
      for i=0 to n
         console.write("*")
      next
      console.writeline()
   end sub
```

```
public overloads sub drawline(byval n as integer
                              , byval ch as char)
    dim I as integer
    console.writeline("third method")
    for i=0 to n
        console.write(ch)
    next
    console.writeline()
  end sub
End class
```

استدعاء الدالة drawline

```
Dim s as new shape()
s.drawline() : s.darwline(10) : s.drawline(15,"-")
```

ملاحظات هامة

ممكن انشاء الطبقة في

- Module و يتم استدعاؤها داخل الـ form.
- نهاية الـ form بعد end form و تستدعى داخل الـ form.
- ملف class داخل نفس الـ project و تستدعى داخل الـ form.

داخل الطبقة المتغير من النوع dim يكافئ private و لابد من انشاء اجراء خاصية له.

Overloading constructors

تعريف اكثر من دالة بناء بنفس الاسم داخل نفس الطبقة مع اختلاف في عدد/نوع/ترتيب المعاملات.

مثال: لدينا طبقة employee تحتوي علي دوال البناء التالية + ...

```
Public Sub New()
        _empID = -1
 End Sub
 Public Sub New(ByVal empID As Integer)
    _empID = empID
       'retrieval of data is hardcoded for testing purposes only.
        If empID = 1 Then
            LoginName = "Smith"
            Password = "js"
            Department = "IS"
            FullName = "Jerry Smith"
        Else
            LoginName = "Jones"
            Password = "mj"
            Department = "HR"
            FullName = "Mary Jones"

       End If
End Sub
```

استدعاء الـ constructors

Dim s as new employee() → 1

Dim e1 as new employee(5) → 2

١- تم استدعاء دالة البناء الاولي new()

٢- تم استدعاء دالة البناء الثانية new(byval empID as integer)

القيم الافتراضية للمعاملات optional parameters

يمكن تحديد معامل او اكثر علي انه optional و في هذه الحالة عند استدعاء البرنامج الفرعي يمكن ارسال قيمة المعامل او عدم ارسالها بشرط اعطاء المعامل قيمة افتراضية عند الانشاء.

المعامل الافتراضي يبدأ من اليمين لليسار.

```
Sub s1 (optional byval st as string = "hi")
    Msgbox(st)
End sub
```

و في هذه الحالة يمكن استدعاء هذا الاجراء بصورتين:

الاولي : عدم ارسال معامل `s1()`

الثانية : ارسال معامل `s1("new value sent")`

مثال

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventA

    ss(10)
    ss(20, 30)
    ss()

End Sub
Public Sub ss(Optional ByVal b As Integer = 5, Optional ByVal a As Integer = 10)
    MsgBox(a & "  " & b)
End Sub
```

مثال

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
    ss(10)
    ss(20, 30)
End Sub
Public Sub ss(ByVal b As Integer, Optional ByVal a As Integer = 10)
    MsgBox(a & "  " & b)
End Sub
```

المهدمات Destructor

عكس ال Constructor، يتم اطلاق هذا الحدث مع انهاء استخدام ال Object، لاصدار صوت Beep مثلاً مع انهاء البرنامج:

```
Protected Overrides Sub Finalize()

End Sub
```

نستفيد من ال Destructor في تنفيذ بعض العمليات قبل تدمير الكائن Object تماماً، احياناً ما نحتاج إلى مسح جميع المتغيرات المرتبطة به في الذاكرة وهو الاستخدام الاشهر لهذا الحدث ايضاً يمكن استخدامه لتغيير العدادات مثلاً والتي تقوم بعد النسخ من فئة معينة.

دالة الهدام Finalize()

- تستدعي آليا لهدم الكائن.

مرحلة الهدم

- المتغيرات في net. لها نوعان (value type , reference type).
- الذاكرة مقسمة الي جزئيين stack , heap.
- Stack منطقة تحجز للتطبيق الذي يعمل حتي يستطيع ان ينفذ و فكرة عملها الداخل آخرا هو الخارج اولا LIFO. و هذا يحدث مع المتغيرات المستخدمة في البرنامج فعند استدعاء الدالة فان كل متغيراتها تدفع الى stack و عند استدعاء هذه الدالة لدالة اخرى فإنها تدفع بمتغيراتها الى stack و هكذا... .
- Heap منطقة تحجز لإنشاء الكائنات التي يتم استخدامها في البرنامج (تخزين الكائنات).

- المتغيرات value type تخزن في stack و عندما ينتهى عملها تحذف.
- المتغيرات reference type تنقسم الى جزئيين (الكائن في الـ heap ، مؤشر الكائن في stack) فعند هدم الكائن يحذف مؤشرة من stack و هناك جامع المهملات الذي يحذف الكائنات من الـ heap التي لا تحمل اى مؤشر في stack.
- دالة الهدم لا نستدعيها بل تنفذ تلقائيا عند انتهاء مدي الرؤيا للكائن.

```
Protected overrides sub finalize()
    mybase.finalize()  //  ليس شرط كتابتها ممكن طباعة رسالة
                           معينة تستدعى عند هدم الكائن
                           او استدعاء الدالة dispose() التي تنظف الاكواد (تحذفها)
End sub
```

Mybase : المقصود بها الفصيلة الام التي نتوارث صفاتها.

المكتبات

المكتبة هي مساحة يخزن (ينشأ) بها العديد من الفصائل يمكن استدعاؤها داخل اى مكان باي مشروع ليتم التعامل مع الفصائل الموجودة بالمكتبة كما لو كانت موجودة بالمشروع.

خطوات انشاء المكتبة:

اولا: انشاء المكتبة

New project ← class library ← نحدد الاسم classlibrary1 ← ok.

تنشأ فصيلة باسم classlibrary1 لا تحتوى على اوامر ممكن تغيير اسمها و لابد من تغيير الاسم في الـ solution.

```
Public class student
   …
End class
Public class employee
   …
End class
```

ثانيا: الترجمة

Build ← build classlibrary1 ← ينشأ ملف .dll داخل الbin

استخدام المكتبة

١) نستدعيها داخل الـ form اولا بربطها بالـ form.

R-click على الـ project في الـ solution ثم add reference للإشارة الى ملف DLL. ثم يظهر مربع حوار add reference و من تبويب browse أو recent نختار اسم الملف classlibrary1.dll ثم open.

٢) نشتق كائن من نوع الفصيلة المراد العمل عليها الموجودة بالمكتبة باستخدام أيا من الطرق التالية:

- Dim A as new classlibrary1.student
- Import classlibrary1.student

```
...
Dim v1 as integer
Dim v2 as string
A.stud-no=125
A.stud-name="Omar"
V1=A.stud-no
...
```

انشاء تطبيق Console

تطبيق بدون واجهات forms عند تنفيذه نحصل على النتيجة في شاشة مستقلة مثل شاشة الـ DOS فينتج ملف module بداخله باسم module1 و به دالة رئيسية sub main() نكتب بها الاوامر المراد تنفيذها عند تشغيل البرنامج.

```
Module module1
  Sub main()
    .....
    MsgBox(".....")
    .....
  End sub
End module
```

اهم الدوال التي تحتويها الفئة Console هي دوال الادخال والاخراج ، وهي:

١. كتابة نص

```
Console.WriteLine(String)
```

٢. قراءة مدخلات من المستخدم

```
Dim x As String = Console.ReadLine()
```

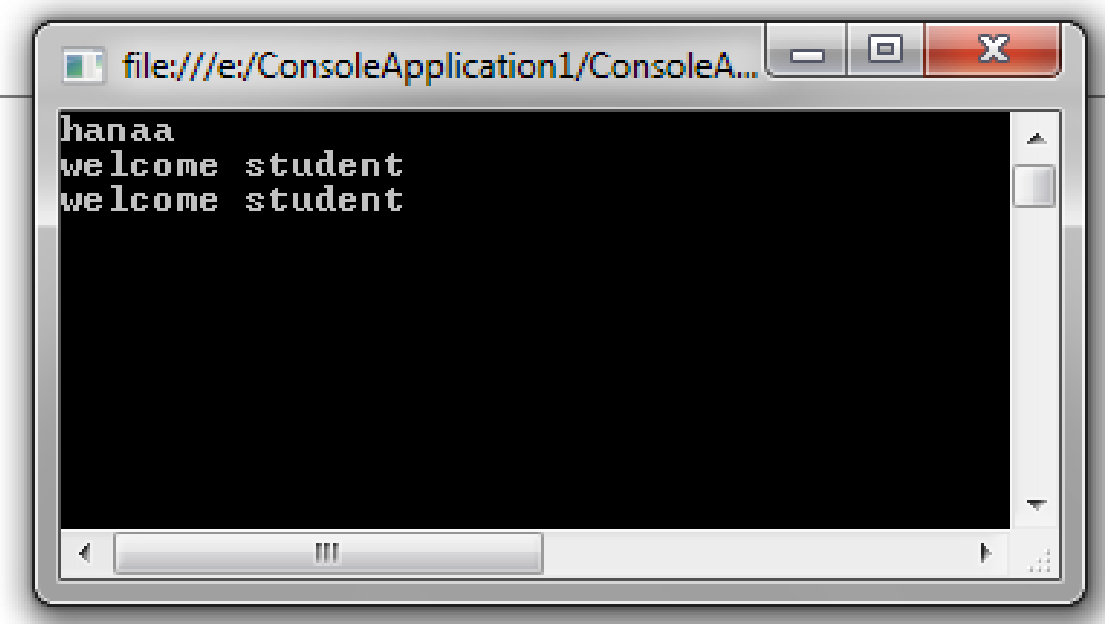
٣. قراءة حرف واحد

```
Console.Readkey()
```

```
Module Module1

    Sub Main()
        Console.WriteLine("hanaa")
        Dim s As String = Console.ReadLine()
        Console.WriteLine(s)
        Console.ReadKey()
    End Sub

End Module
```



المصفوفات ذات البعدين

```
Dim  x(3,4)  as  integer
```

حجز مصفوفة x مكونة من ٣ صفوف ، ٤ اعمدة اذا كان التخزين يبدأ من الموقع ١ ، أو حجز مصفوفة x مكونة من ٤ صفوف ، ٥ اعمدة اذا كان التخزين يبدأ من الموقع صفر.

```
Dim  c , r
Dim  A(3,4) as  integer
For  r=1 to 3
   for  c=1 to 4
        A(r,c) = inputbox()
   next  c
Next  r
```