

ចេតនា ១

ការណែនាំ Introduction

I. ការណែនាំ Microsoft Visual Basic.Net 2008

នៅក្នុងខែ មិថុនា ឆ្នាំ ២០០០ ក្រុមហ៊ុន Microsoft បានផ្សព្វផ្សាយពី *.Net initiative* ដែលជាជំនាន់ថ្មីរបស់ VB សម្រាប់ប្រើជាមួយ Internet, Web និង ការបង្កើត Software ។ គោលបំណងរបស់ .Net គឺ មានភាពងាយស្រួលសម្រាប់អ្នកអភិវឌ្ឍ ដោយមានភាសា និង Platform ច្បាស់លាស់។ VB.net វាអាចជួយដល់អ្នកបង្កើត ក្នុងការធ្វើគំរោង Software ដោយប្រើប្រាស់ ភាសារបស់វាដូចជា (Visual Basic.Net, Visual C++.Net, C#.Net) និង ភាសាផ្សេងៗទៀត ហើយអាចប្រើបច្ចេកទេស *Active Server Page(ASP).Net* របស់ Microsoft ពី ប្រព័ន្ធបង្កើត Application របស់ Web បាន។

II. ការស្វែងយល់ពី .Net Framework និង Common Language Runtime

Microsoft *.Net Framework* គឺជាបេះដូងនៃ *.Net strategy* ។ Framework ប្រើសម្រាប់គ្រប់គ្រង និងដំណើរការ application នឹង Web service វាមាន class library (ដែលគេហៅថា *Framework class library or FCL*) ដែលផ្តុកទៅដោយបណ្តុំនៃ Function Packages ។

Common Language Runtime(CLR) គឺកន្លែងនៃ .Net Framework វាអាចដំណើរការកម្មវិធី VB.Net ។ កម្មវិធីត្រូវបាន compile ទៅជា *Microsoft Intermediate Language(MSIL)* ។ កូដដែលបាន converted ទៅជា MSIL បន្ទាប់មក compiler ផ្សេងទៀតមកបកប្រែ MSIL ទៅជាភាសាមាស៊ីនកូដ។

ហេតុអ្វីត្រូវការបកប្រែមកជាមួយទៀតដែលបំប្លែងពី Visual Basic.Net ទៅជា MSIL ជំនួសការបកប្រែដោយផ្ទាល់ពី Visual Basic.Net ទៅជាមាស៊ីនកូដ? ហេតុផលសំខាន់គឺ MSIL អាចដំណើរការលើ Operating System ផ្សេងៗបាន ហើយវាអាចប្រតិបត្តិ (interoperability) រវាងភាសា និងការគ្រប់គ្រងប្រតិបត្តិ ដូចជាការគ្រប់គ្រង memory និង security ។

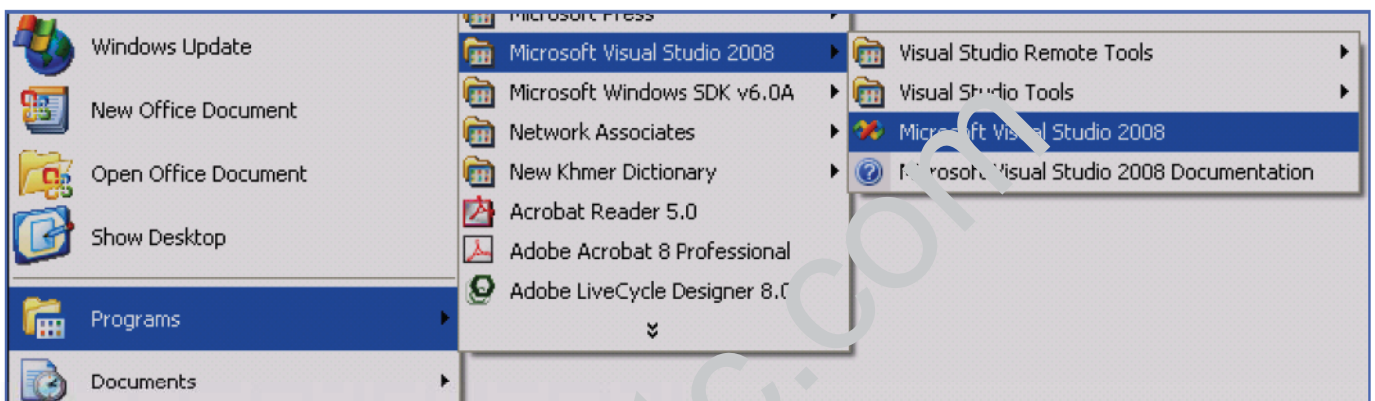
.NET Framework ក៏បានផ្តល់ដល់អ្នកសរសេរកម្មវិធីជាមួយនិង library ដ៏ធំនៃ Classes ។ Library នេះត្រូវបានគេហៅថា *Framework Class Library (FCL)* ដែលប្រើដោយភាសា .Net ផ្សេងៗ ។

III. ការចាប់ផ្តើម Visual Studio 2008

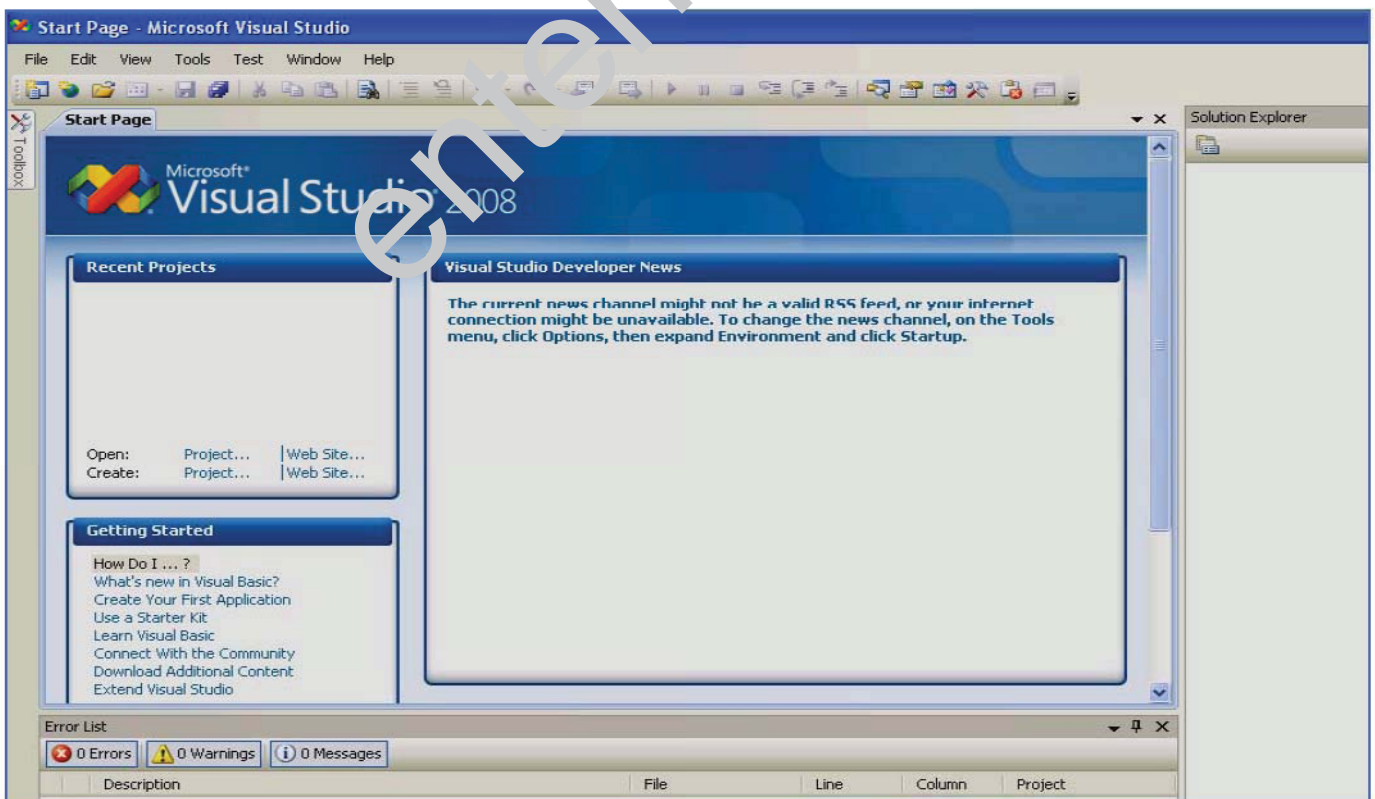
យើងអាចបើកកម្មវិធី Visual Basic ដំណាក់កាលដំបូងខាងក្រោម:

- ៣. ចុច Start
- ៤. ច្រើសរើសយក Programs
- ៥. ច្រើសរើស Microsoft Visual Studio 2008
- ៦. ច្រើសរើស Microsoft Visual Studio 2008

ដូចរូបខាងក្រោម:



ពេលនោះអ្នកនឹងទទួលដូចរូបខាងក្រោម:



IV. ការចាប់ផ្តើមកម្មវិធី Visual Basic .NET

អ្នកអាចបើកកម្មវិធី Visual Basic ដូចខាងក្រោម៖

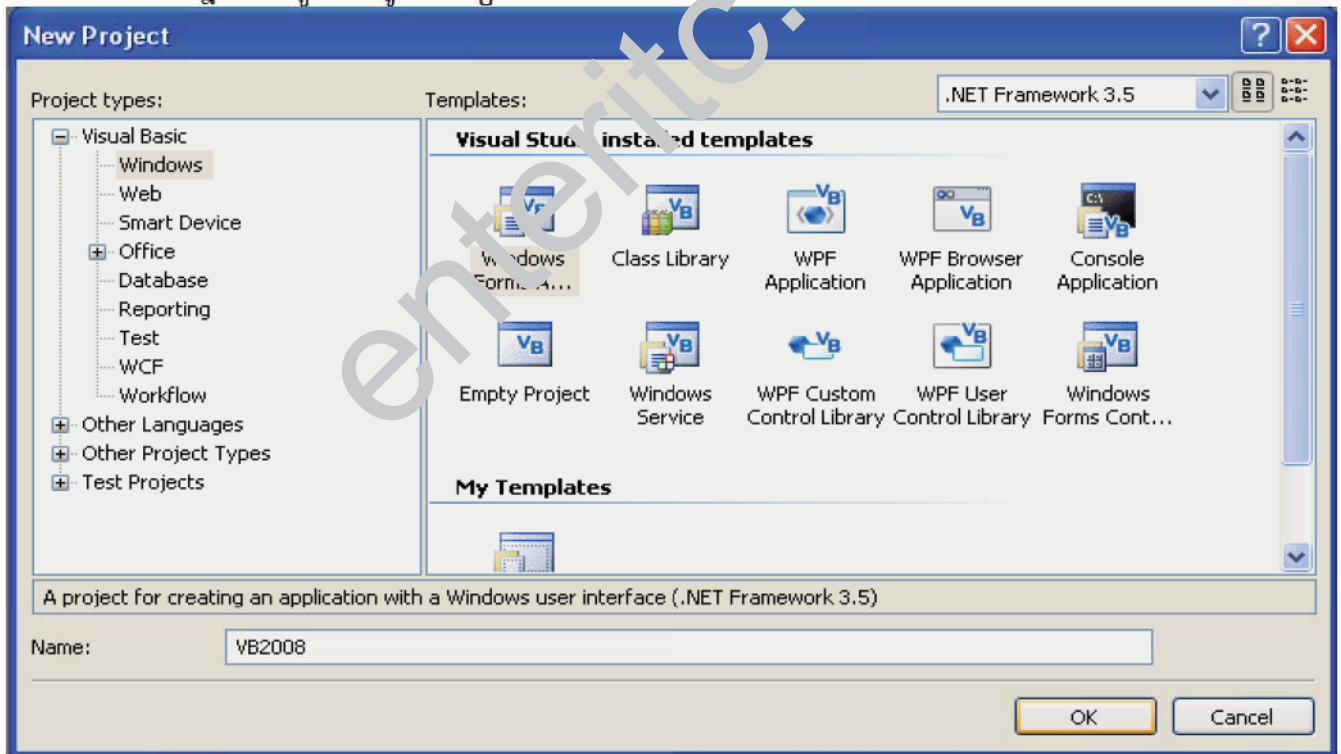
១-ជ្រើសរើសយក File

២-ជ្រើសរើសយក New Project (ឬសម្រាប់ Ctrl+N)

ដូចរូបខាងក្រោម៖



ពេលនោះអ្នកនឹងទទួលបានដូចខាងក្រោម៖



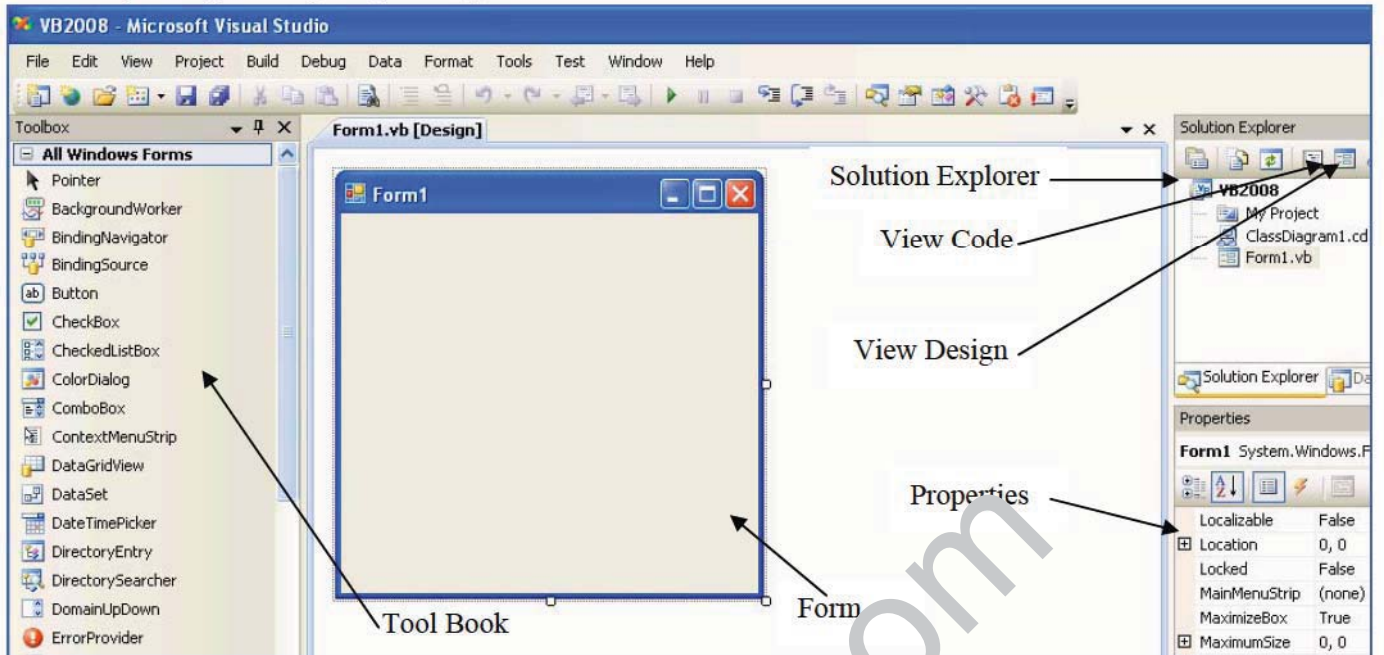
៣-ជ្រើសរើសយក Visual Basic

៤-បន្ទាប់មកជ្រើសរើសយក Windows Application

៥-ចុច OK

ចំណាំ៖ នៅត្រង់ Name គឺជាឈ្មោះ Folder ដែលប្រើសម្រាប់គ្រប់គ្រង Project វាមានទីតាំងដូចខាងក្រោម៖

សូមពិនិត្យមើលលទ្ធផលដូចខាងក្រោម៖



- Toolbox គឺជាឧបករណ៍ប្រើសម្រាប់ Design application ។
- Form គឺជាផ្ទាំងប្រើសម្រាប់ Design application ។
- Solution Explorer ប្រើសម្រាប់បង្ហាញការងារនៅក្នុង Project ។
- View Code ជា Button មួយសំរាប់អោយលោកអ្នកចូលទៅដល់កន្លែង សរសេរ Code របស់កម្មវិធី VB.Net ។
- View Design ជា Button មួយសំរាប់ អោយលោកអ្នកចូលទៅដល់កន្លែង Form Design
- Properties ប្រើសម្រាប់កំណត់លក្ខណៈអោយ controls ដោយមិនចាំបាច់សរសេរ code ។

V. របៀបដំណើរការកម្មវិធី (Start Debugging)

ឧបមាថា យើងមានទម្រង់ Form ដូចខាងក្រោម :



Source Code:

```

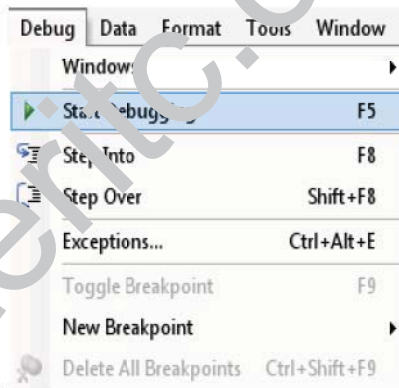
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim Language As String

    Language = Me.TextBox1.Text
    Language = Language.ToUpper

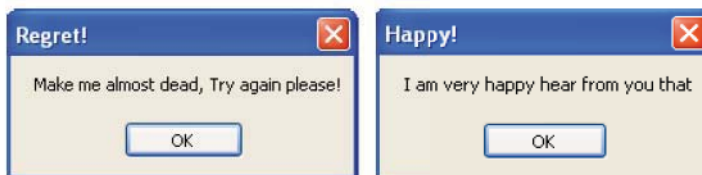
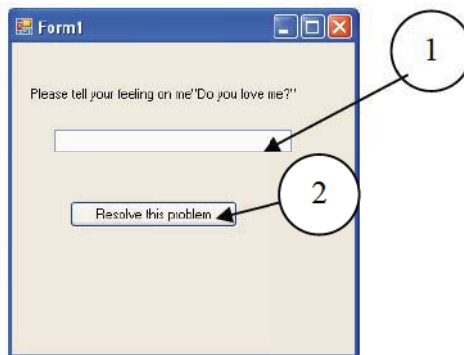
    If Language = "OK" Or Language = "YES" Then
        MsgBox("I am very happy hear from you that ")
    End
    Else
        MsgBox("Make me almost dead, Try again please!")
        Me.TextBox1.Focus()
        Me.TextBox1.SelectAll()
    End If
End Sub

```

ដើម្បីដំណើរការកម្មវិធី បន្ទាប់អ្នកបានសរសេរហើយ សូមលោកអ្នកចុចលើ Debug Menu → Start Debugging ឬ លោកអ្នកអាចចុច Short Cut Key ដោយចុចលើ Key F5 ។ ឡើយទៅក៏យើងអាចចុចលើសញ្ញា Short Cut ដែលមាននៅលើ Toolbar ដូចរូបខាងក្រោម :



បន្ទាប់ពីលោកអ្នកបានដំណើរការកម្មវិធីដូចមកនោះលោកអ្នកនឹងទទួលបានទម្រង់ Application ដែលលោកអ្នកបានសរសេរដូចរូបខាងក្រោម :



VI. អប្បបរមាបញ្ចប់ដំណើរការកម្មវិធី (Stop Debugging)

ដើម្បីបញ្ចប់ដំណើរការកម្មវិធីដែលកំពុងដំណើរការ យើងត្រូវចុចលើ Debug Menu => Stop Debugging ឬ ចុចលើរូបភាព Short Cut លើ Toolbar



VII. ការរក្សាទុកសំណាក

នៅពេលការងារត្រូវបានបញ្ចប់ នោះយើងត្រូវចំណងកសារទុកនៅក្នុង ព័ត៌មាន

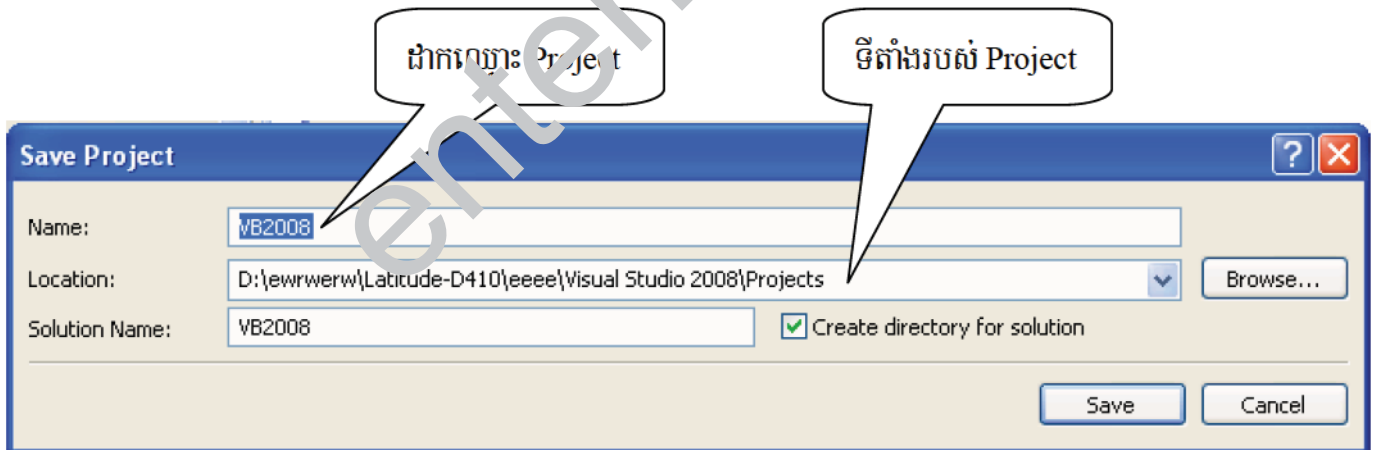
តាមរបៀបដូចខាងក្រោម:

១- ជ្រើសរើសយក File Menu

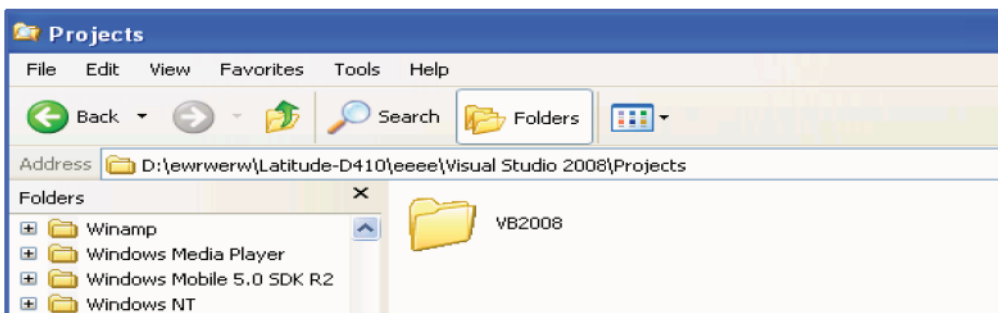
២- ជ្រើសរើសយក Save All (Ctrl+Shift+S)

ចំណាំ: Save All សម្រាប់រក្សាទុកការងារទាំង ទុក្ខយ នៅក្នុង Project ។

ចំណែកឯ Save FormName.vb វិញ ត្រូវសម្រាប់រក្សាទុកតែការងារដែលកំពុងធ្វើការប៉ុណ្ណោះ ។



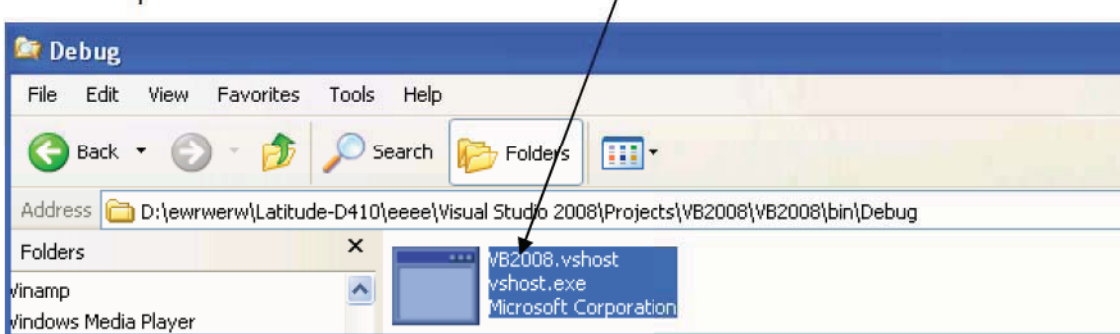
បន្ទាប់ពីយើងចុច Save Button រួចមកនោះលោកអ្នកនឹងទទួលបាន Project File ដូចរូបខាងក្រោម:



ចំណាំ :

បន្ទាប់ពីពេលដែលយើងបាន Save ហើយដំណើរការកម្មវិធី (Start Debugging) នោះលោកអ្នកនឹងទទួលបាន .EXE

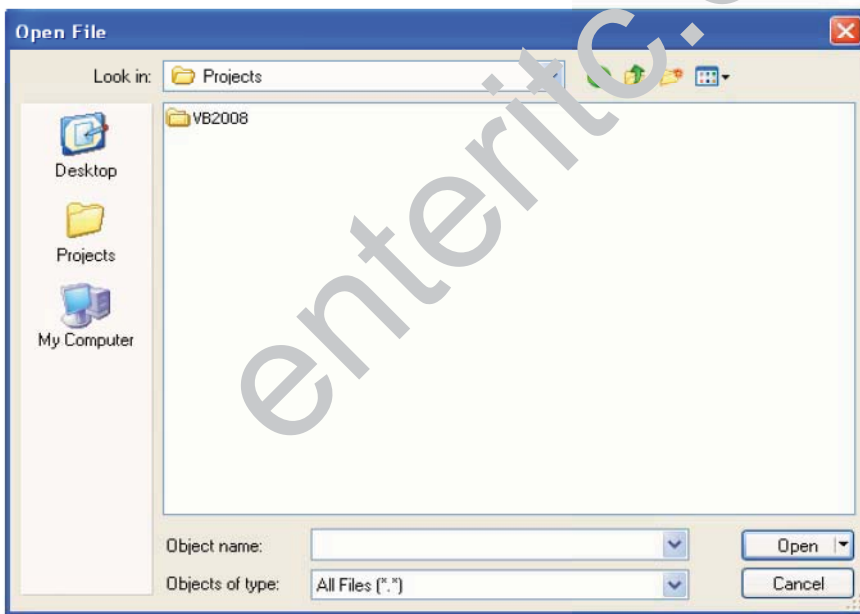
File ដែលនៅក្នុង Folder ដូចរូបខាងក្រោម :



VIII. ការបើកយកការងារបកប្រែកម្មវិធី

នៅពេលដែលអ្នកចង់បើកឯកសារនៅក្នុង Project មកធ្វើការ នៅមកអ្នកធ្វើវាតាមរបៀបដូចខាងក្រោម :

- ១- ជ្រើសរើសយក File Menu
- ២- ជ្រើសរើសយក Open Project (Ctrl+O)

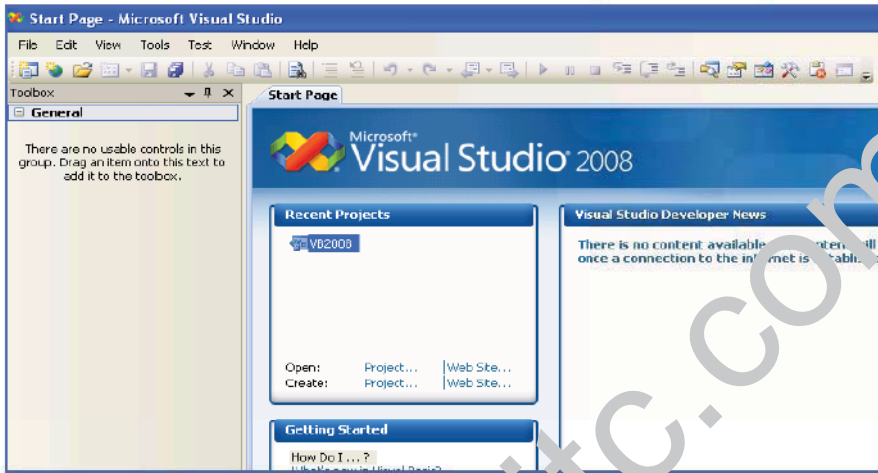


- ៣- ជ្រើសរើសយក Project
- ៤- ចុច Button Open
- (អ្នកអាចបើកវាតាម File menu=>Open File...)

IX. របៀបបើក Project ដែលមានស្រាប់ដើម្បីកែ

ដើម្បីធ្វើការកែប្រែលើកម្មវិធីដែលយើងបានបង្កើតហើយ យើងអាចធ្វើតាមវិធីច្រើនយ៉ាងដូចជា :

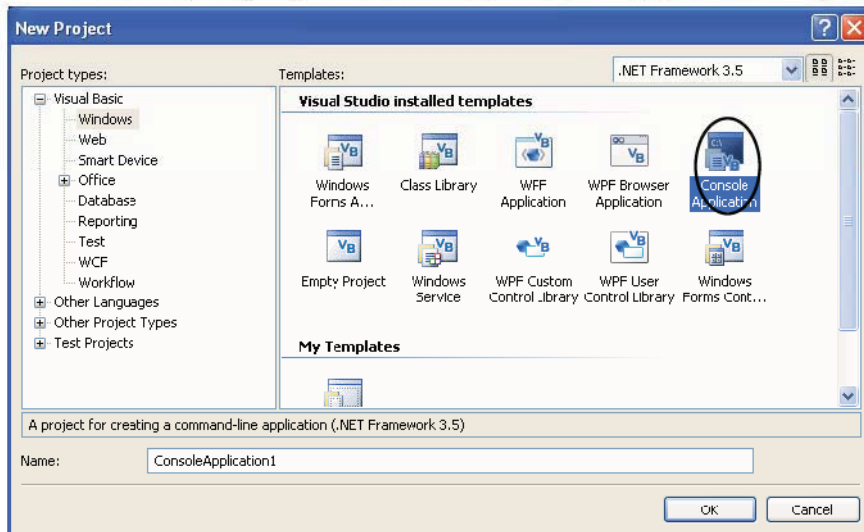
- ចូលទៅ Double Click លើ ឈ្មោះកម្មវិធី (Project) ដែលមាន Extension *.sln
- បើកកម្មវិធី Microsoft Visual Studio.NET => ចុចលើ File Menu => Open File => បន្ទាប់មកសូមរកមើល ឈ្មោះ Project File ដែលមាន Extention *.sln ហើយចុចលើវា បន្ទាប់មកសូមចុចលើ Open Button ។
- ប្រសិនបើលោកអ្នកចង់បើក Project ដែលលោកអ្នក ទើបតែបិទធ្វើនោះលោកអ្នក អាចអនុវត្តដូចរូបខាងក្រោម :



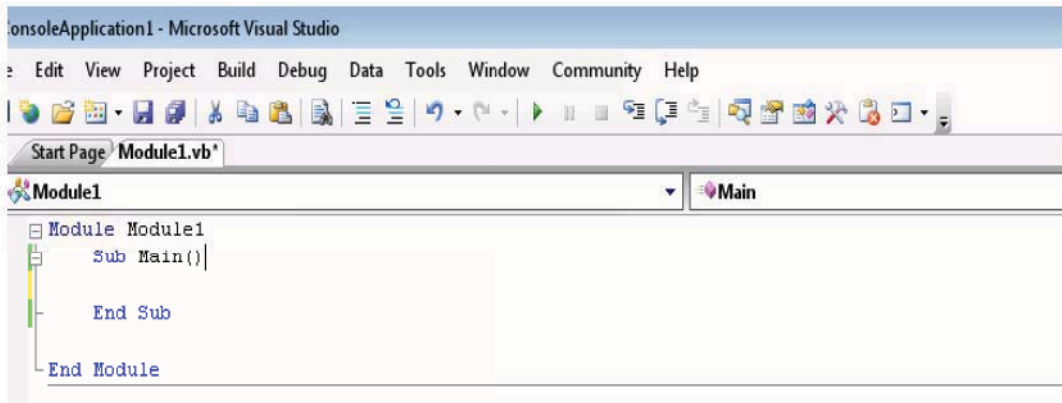
X. របៀបបង្កើត Console Application

នៅក្នុង Visual Basic 2008 យើងអាចបង្កើត បង្កើត ដែលអាចដំណើរការ នៅលើ Command Prompt Window បាន ។ Command Prompt window គឺជា DOS windows នោះទេ ។ ប្រសិនបើលោកអ្នកមើលរូបភាពខាងលើវា គឺវា មានលក្ខណៈស្រដៀងគ្នាទៅនឹង DOS ដើម្បីបង្កើត Console Application លោកអ្នកអាចអនុវត្តតាមជំហានខាងក្រោម:

- ចុច File Menu => New Project រួចជ្រើសរើសយកពាក្យ Console Application=> ចុចលើ OK Button ។



បន្ទាប់មកលោកអ្នកនឹងទៅដល់កន្លែងសរសេរកូដ ដូចរូបភាពខាងក្រោម :



ចំណាំ :

ចំពោះ Console Application មិនមាន User Interface ដូចជា window Application នោះទេ គឺវាមានតែ Module ដូចរូបភាពខាងលើប៉ុណ្ណោះ ។

ឧទាហរណ៍:

ឧបមាថា លោកអ្នកចង់សរសេរកម្មវិធីបង្ហាញសញ្ញាផ្កាយនៅលើ Console Application ដែលមានកំរូដូចរូបភាព ត្រូវអនុវត្តតាម Source Code ដូចខាងក្រោម :

Source Code:

Module Module1

```

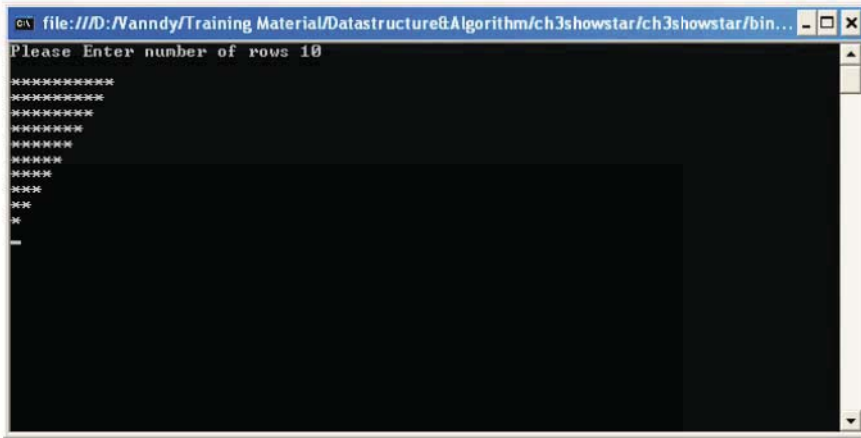
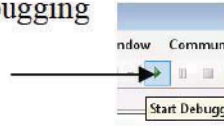
Dim n As Integer
Sub Main()
    Console.WriteLine("Please Enter number of rows ")
    n = Console.ReadLine
    display(n)
    Console.ReadKey()
End Sub
Public Function display(ByVal ss As Integer)
    If ss < 0 Then
        Return 0
    Else
        Console.WriteLine(vbNewLine)
        write(ss)
        display(ss - 1)
    End If
    Return 0
End Function
Public Function write(ByVal p As Integer)
    If p < 0 Or p = 0 Then
        Return 0
    Else
        Console.WriteLine("*")
        write(p - 1)
    End If
    Return 0
End Function

```

End Module

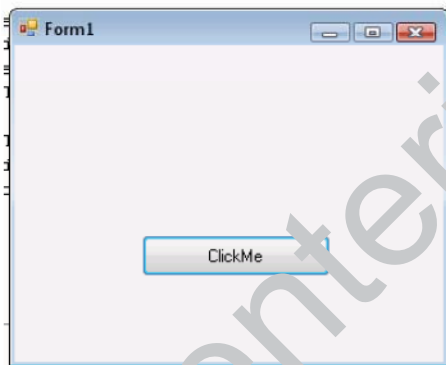
-បន្ទាប់ពីយើងសរសេររចបំ យើងចាប់ផ្តើម Start Debugging

លទ្ធផល :



លំហាត់អនុវត្ត

1/ ចូរបង្កើតនូវ Project មួយដែលមានឈ្មោះថា MyFirstApp បន្ទាប់មកបង្កើត Form ដូចរូបទី ១



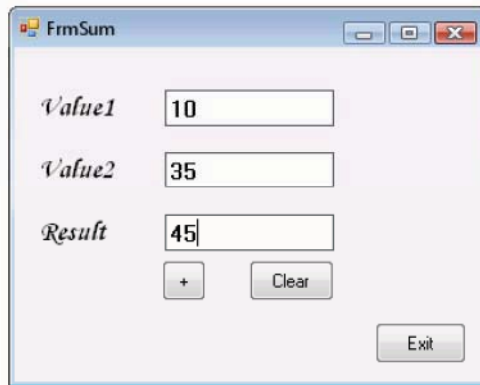
Source Code:

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

```
    If Me.btnclick.Text = "ClickMe" Then
        Me.lblshow.Visible = True
        Me.lblshow.Text = "Welcome to all lecturers and professors at NUM"
        Me.btnclick.Text = "HideMe"
    Else
        Me.btnclick.Text = "ClickMe"
        Me.lblshow.Visible = False
        Me.lblshow.ForeColor = Color.Red
    End If
```

```
End Sub
```


2/ ចូរបង្កើត Form មួយសំរាប់ធ្វើផលបូកនៃពីរចំនួនរវាងតំលៃលេខដែលមាននៅក្នុង Textbox1 និង Textbox2 ទរង់ Form ដូចខាងក្រោម:



Source Code:

```

Private Sub btnsum_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnsum.Click
    If Me.txt1.Text = "" Then
        MessageBox.Show("Please Enter Number in Textbox1")
    ElseIf Me.txt2.Text = "" Then
        MsgBox("Please Enter Number in Textobox2")
    Else
        Me.txtresult.Text = Cdbl(Me.txt1.Text) + Cdbl(Me.txt2.Text)
    End If
End Sub

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    Me.txt1.Clear()
    Me.txt2.Text = ""
    Me.txtresult.Clear()
    Me.txt1.Focus()
End Sub

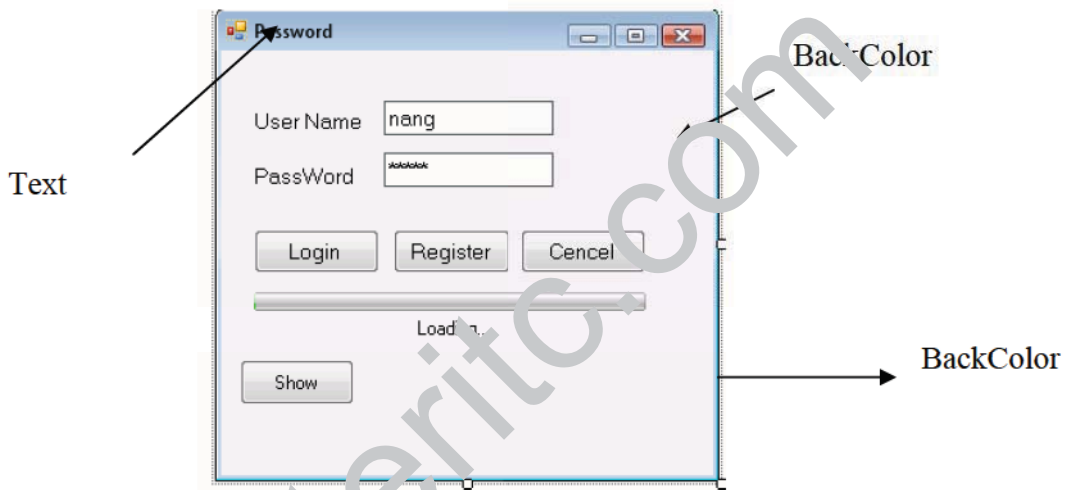
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnexit.Click
    If MsgBox("Are you want to exit?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then
        End
    Else
        End If
End Sub
    
```

បេឡេនី ២

ការតុបតែង *Interface* ជាមួយ *Form*

I. Form

នៅក្នុងភាសា Visual Basic បានគេហៅថាជា Container ដែលត្រូវប្រើសំរាប់ផ្ទុករាល់ Controls ដើមបង្កើត User Interface ។ រាល់ Form ដែលយើងបានឃើញគេហៅថា Form Object ។



II. របៀបតុបតែង Form

ដើម្បីតុបតែង Form យើងត្រូវ រាល់ ៤ ចំណុចខាងក្រោមជាមុនសិន :

1. **Object:** គឺសំដៅទៅលើអ្វីមួយដែលមាន Properties ឬ Methods ឬ Events ។ Objects នៅក្នុង VB មានដូចជា : Form, Module, Class, TextBox, Label, ListBox, Combobox, TreeView, ListView, MenuStrip, ContextMenuStrip, StatusStrip... ។

2. **Property:** គឺជាលក្ខណៈរបស់ Object ។ មានន័យថាយើងអាចធ្វើការផ្លាស់ប្តូរលក្ខណៈរបស់ Object អ្វីមួយ បានតាមរយៈ Properties របស់វា ។ ដើម្បីមើលអោយឃើញថាតើ Object អ្វីមួយរបស់វាមានអ្វីខ្លះ សូមយើងចុច Mouse ខាងស្តាំអោយចំ Object ហើយជ្រើសរើសយក Properties នោះវានឹងបង្ហាញ Form Properties សំខាន់ៗ ។

3. **Methods:** គឺសំដៅទៅលើ Methods របស់ Object ថាតើ Object ខ្លួនឯងអាចធ្វើអ្វីបាន ។

ឧទាហរណ៍ :

FormName.Show
FormName.Close

... គឺជា Method របស់ Form Object ។

4. Events: គឺសំដៅទៅលើការបញ្ជាអោយ Object ធ្វើអ្វីមួយ ហើយធ្វើនៅពេលណា។

ឧទាហរណ៍:

- Form_Load(...)
- ObjectName_MouseDown
- ObjectName_MouseUp(...)
- ObjectName_MouseLeave(...)
- ObjectName_MouseOver(...)
- ObjectName_MouseMove(...)
- ObjectName_Click(...)
- ObjectName_KeyDown(...)
- ObjectName_Keyup(...)
- ObjectName_KeyPress(...)
- ObjectName_TextChang(...)

ព្រមទាំង Event... ជាច្រើនទៀត

III. របៀបបង្កើត Form មួយទៀត

ក្នុងការបង្កើត Application មួយតែងតែមាន Process មួយៗជាច្រើន ហើយ Process នីមួយៗក៏ទាមទារអោយមានទំនាក់ទំនងគ្នា។ ដូច្នេះទាមទារអោយយើង ធ្វើការបង្កើតទំរង់នៃ Interface សំរាប់ដំណើរការនៃ Process ។ ដូចនេះយើងត្រូវបង្កើត Window Form លើសពីមួយ។ នៅក្នុងកម្មវិធី VB.Net 2005 គឺផ្តល់អោយយើងបង្កើត Windows Form មួយទំនងជា :

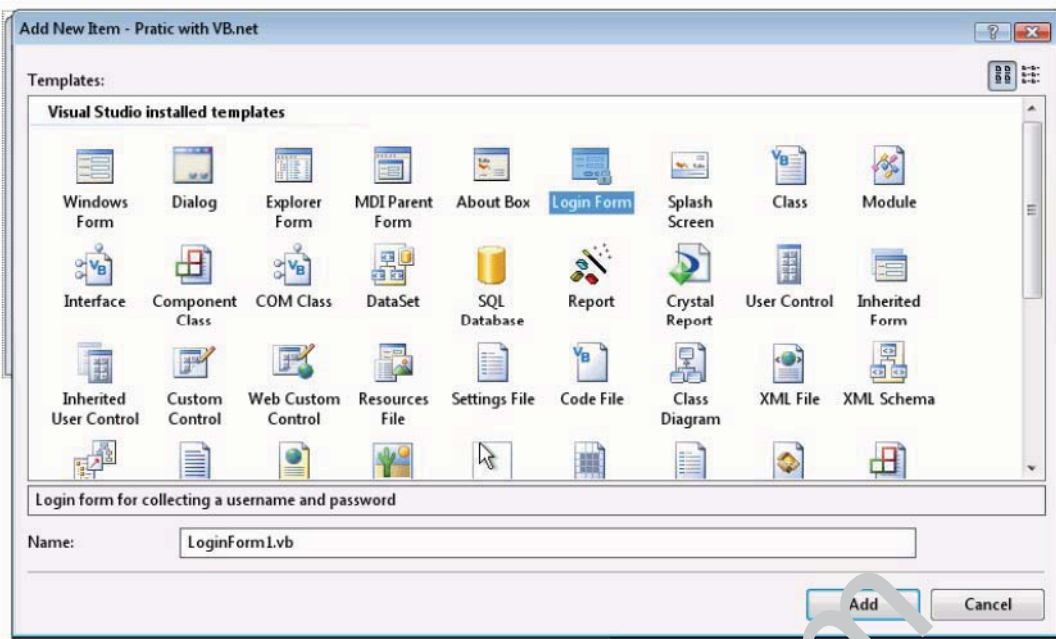
- ☞ Window Form
- ☞ Login Form
- ☞ Explorer Form
- ☞ MDI Present Form...

a. ការបង្កើត Window Form

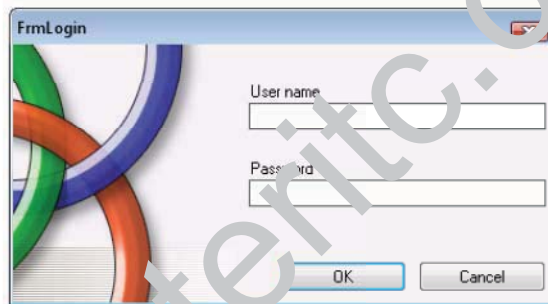
- ចុចលើ Project Menu=> AddWindowForm
- Select យក Window Form ហើយដាក់ឈ្មោះអោយ Form
- បន្ទាប់មកចុចលើ Add Button

b. ការបង្កើត Login Form

Login Form គឺជា Form មួយដែលលោកអ្នកប្រើសំរាប់ ធ្វើប្រព័ន្ធសុវត្ថិភាពពេលចូលប្រើកម្មវិធី ដោយ ក្នុង Visual Basic Studio .Net ផ្តល់អោយលោកអ្នកបង្កើត Interface សំរាប់ប្រព័ន្ធ Login System ដោយស្វ័យប្រវត្តិ។ ការបង្កើតវាស្រដៀងគ្នាទៅនឹង ការបង្កើត Window Form ដែរគ្រាន់តែយើងជ្រើសរើសយក Login Form Object ប៉ុណ្ណោះ។



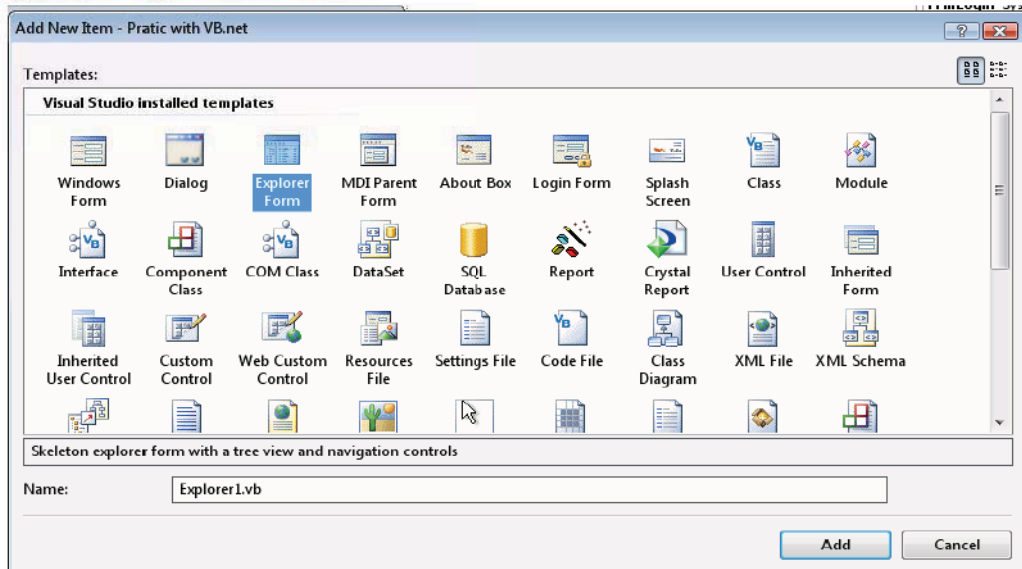
បន្ទាប់មកយើងនឹងទទួលបាន Form(Form Login) ដូចរូបខាងក្រោម:



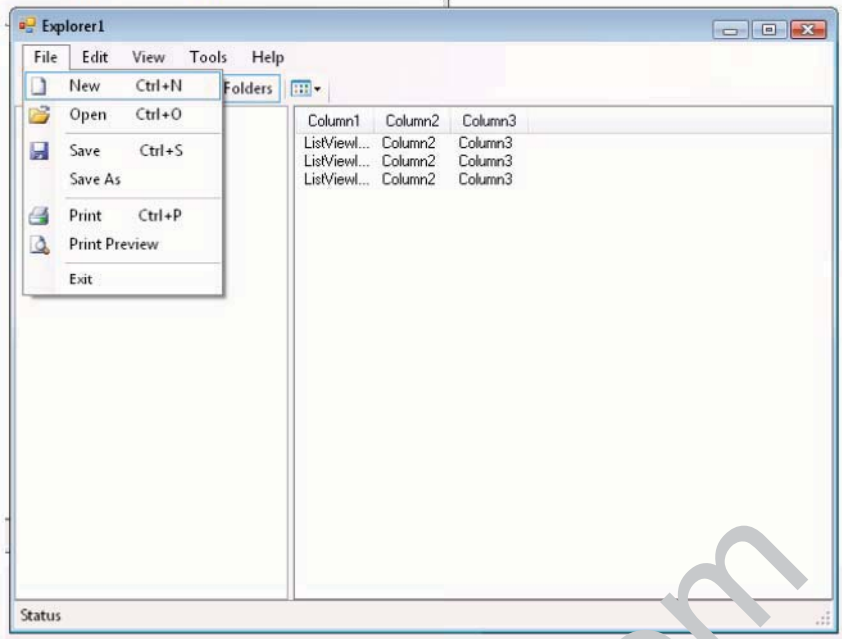
c. របៀបបង្កើត Explorer Form

Explorer Form គឺជាទម្រង់ Form មួយដែលមានលក្ខណៈជាទម្រង់ Window Explorer រាប់លំដាប់ Widnow ។

ចំពោះការបង្កើតដូចគ្នានឹងការបង្កើត Form ខាងលើ ។

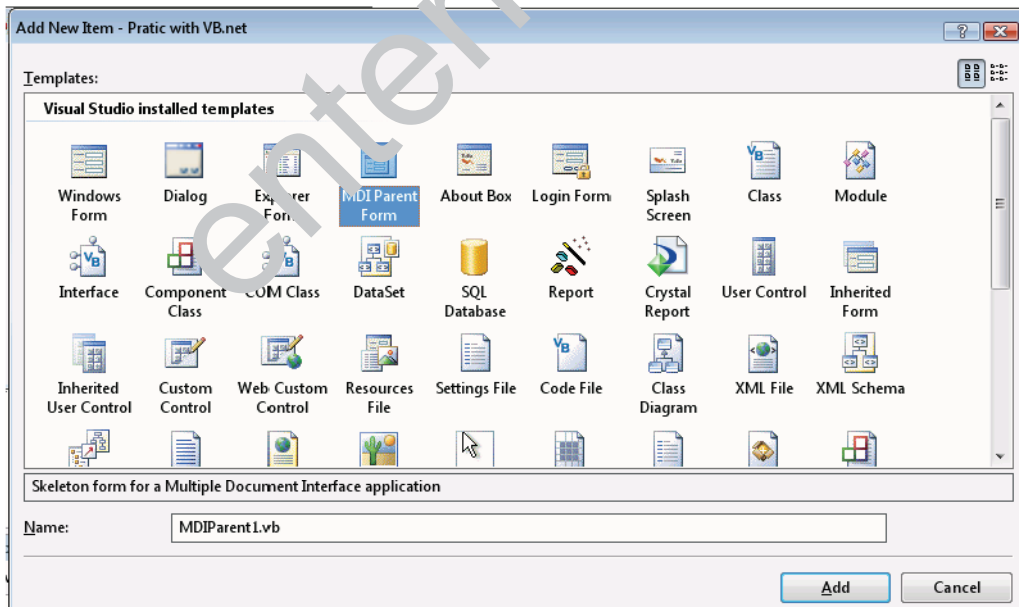


បន្ទាប់មកយើងនឹងទទួលបានដូចរូបខាងក្រោម:

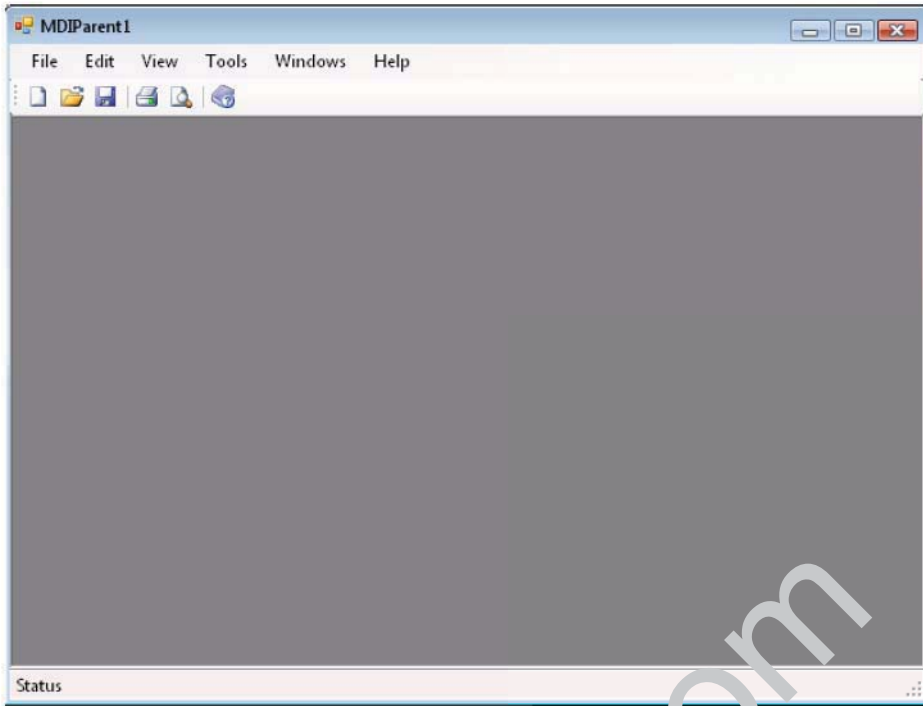


d. ការបង្កើត MDI(Multiple Document Interface) Form

MDI Form គឺជា Form មេ (Main Form) ប្រើសំរាប់បង្ហាញអោយឃើញជាច្រើន ហើយនៅក្នុង Form នេះមានផ្ទុក Menubar, Tool bar, Status bar សំរាប់អោយលោកអ្នក Link ទៅបាន Process ផ្សេងៗដែលជា Functions មាននៅក្នុងប្រព័ន្ធកុំព្យូទ័រ។ ការបង្កើតដូចជាមានមុនដែរ គ្រាន់តែប្រើសរសេរពាក្យថា MDI Parent Form

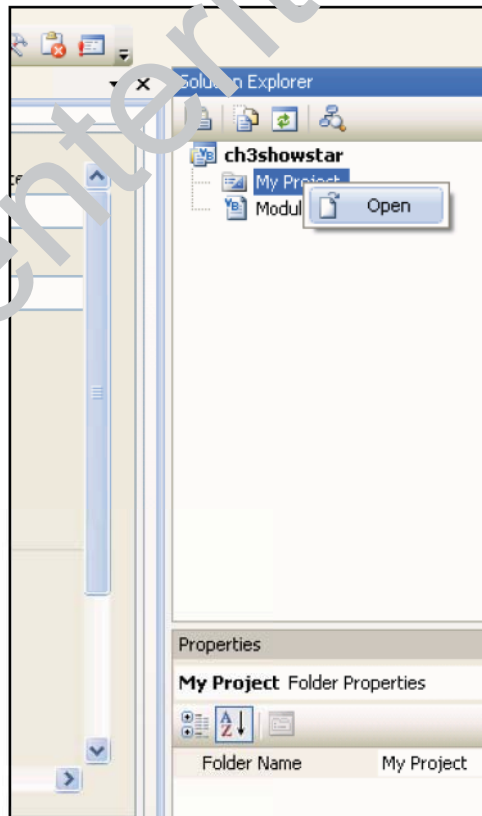


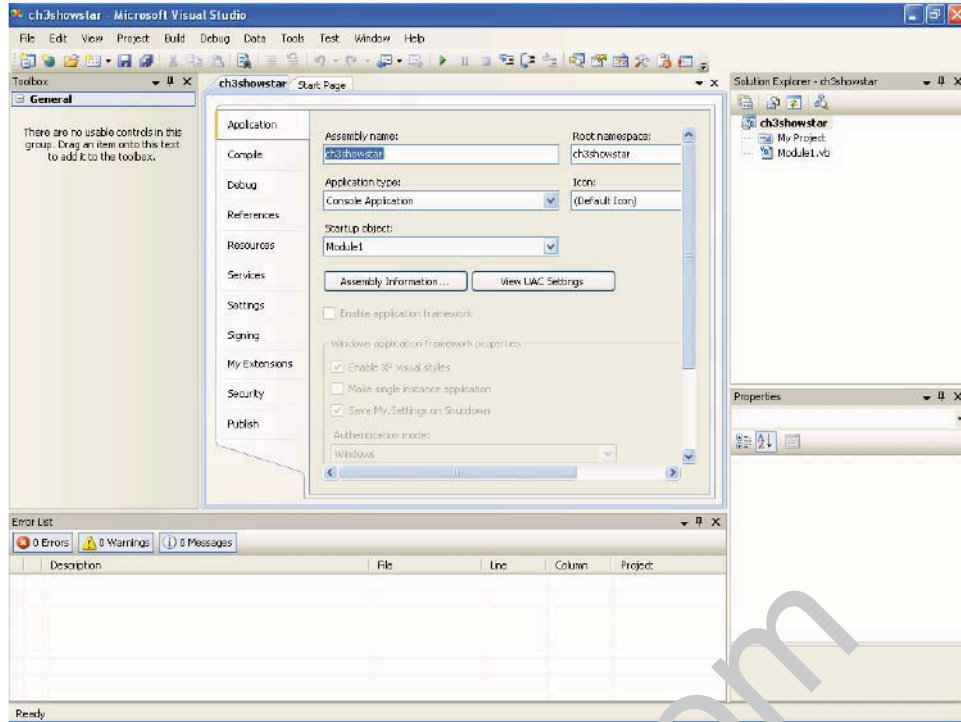
បន្ទាប់មកយើងទទួលបានដូចរូបខាងក្រោម:



IV. របៀបប្រើសរសេរ Form ដើម្បីដំណើរការមុន

- ប៊ុច Mouse ខាងស្តាំអោយចំណេញ៖ My Project ដែលមាននៅក្នុង Project Explorer ហើយប៊ុច Open





V. អប្បបរមាអោយបើក Form ផ្សេងទៀត

a. Syntax ទី១

FormName.Show()

ចំពោះ Syntax នោះមានន័យជាបង្គាប់ឱ្យបើក Form មួយហើយ Form ដែលនៅខាងក្រោយយើងអាចយក Mouse ទៅដាក់ Focus នៅលើ Form ដែលយើងមើលឃើញដំបូងនោះទេ ។

ឧទាហរណ៍:

Source Code:

```

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

```

b. Syntax ទី២

FormName.ShowDialog()

ចំពោះ Syntax ទី២ នេះមានន័យថា បង្គាប់ឱ្យបើក Form មួយហើយ Form ដែលនៅខាងក្រោយយើងមួយទៀតមិនអាចយក Mouse ទៅដាក់ Focus នៅលើ Form នេះបានទេ ពេលដែលយើងមើលឃើញ Form មួយដំបូង ។

VI. អប្បបរមា បិទ Form

Syntax :

FormName.Close()
Or

Me.Close()

Source Code:

ឧទាហរណ៍:

```

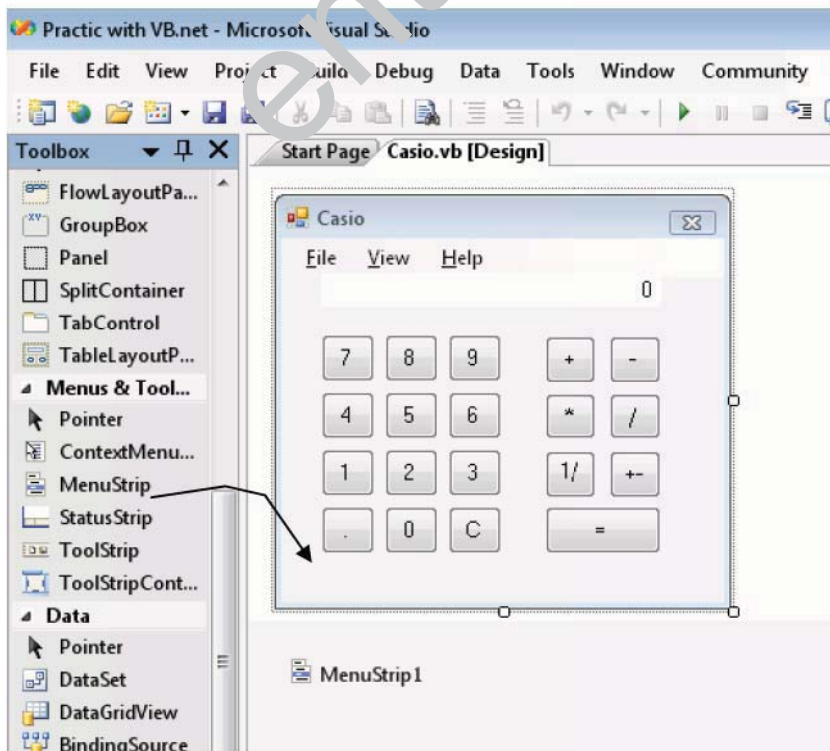
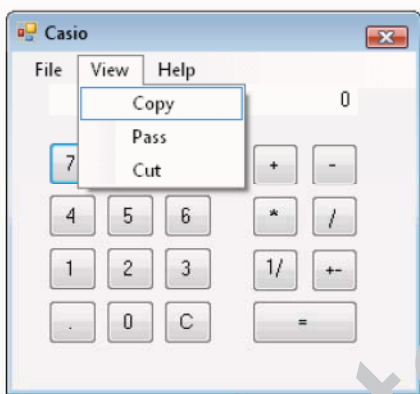
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    If MsgBox("Are you want to exit?", MsgBoxStyle.YesNo) = MsgBoxResult.Yes Then
        Me.Close()
    End If
End Sub

```

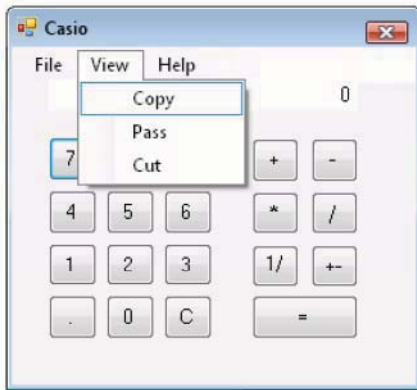
VII. របៀបបង្កើត Menubar

ដើម្បីបង្កើត Menubar យើងត្រូវរៀបចំ Control ដែលមានឈ្មោះថា MenuStrip ។

ឧទាហរណ៍:



-បន្ទាប់មកយើងនឹងទទួលបាន Form ដូចរូបខាងក្រោម:



Source Code:

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

```
Me.lbldisplay.Text = ""
```

```
End Sub
```

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

```
Clipboard.SetText(Me.lbldisplay.Text)
```

```
End Sub
```

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

```
Me.lbldisplay.Text = Clipboard.GetText()
```

```
End Sub
```

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

```
Clipboard.SetText(Me.lbldisplay.Text)
```

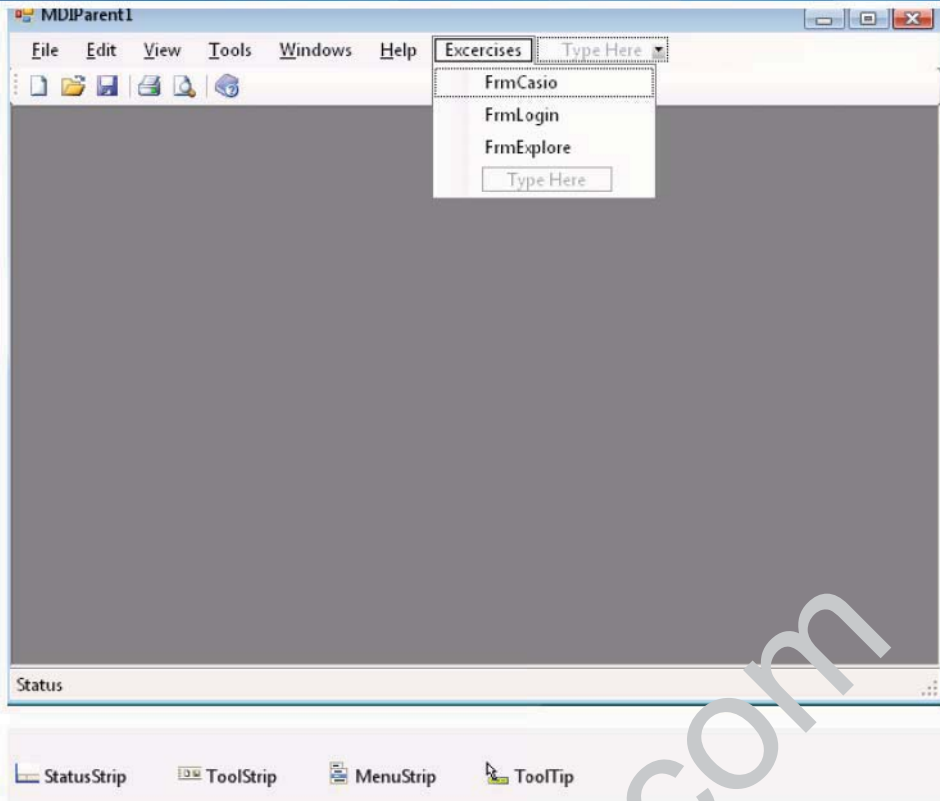
```
Me.lbldisplay.Text = ""
```

```
End Sub
```

VIII. របៀបបង្កើត MDI Parent And Child Form

ដើម្បីបង្កើត Child Form យើងអនុវត្តដូចរូបខាងក្រោម:

- ចូលទៅជ្រើសរើស FrmMain Form ដើម្បីអោយវាដំណើរការមុនគេ
- បន្ទាប់មកបន្ថែម Menu អោយវាដូចរូបខាងក្រោម:



Objects	ឈ្មោះវត្ថុអោយ Object
FrmCasio	SubMnuCasio
FrmLogin	SubMnuLogin
FrmExplore	FrmSubExplor

ពេលបើក MDI Parent Form បន្ទាប់មកយើងបើក Exercises ហើយជ្រើសរើស Form ដែលយើងពេញចិត្តនោះវានឹង

បើកអោយយើងឃើញ។ ការសរសេរកូដ៖

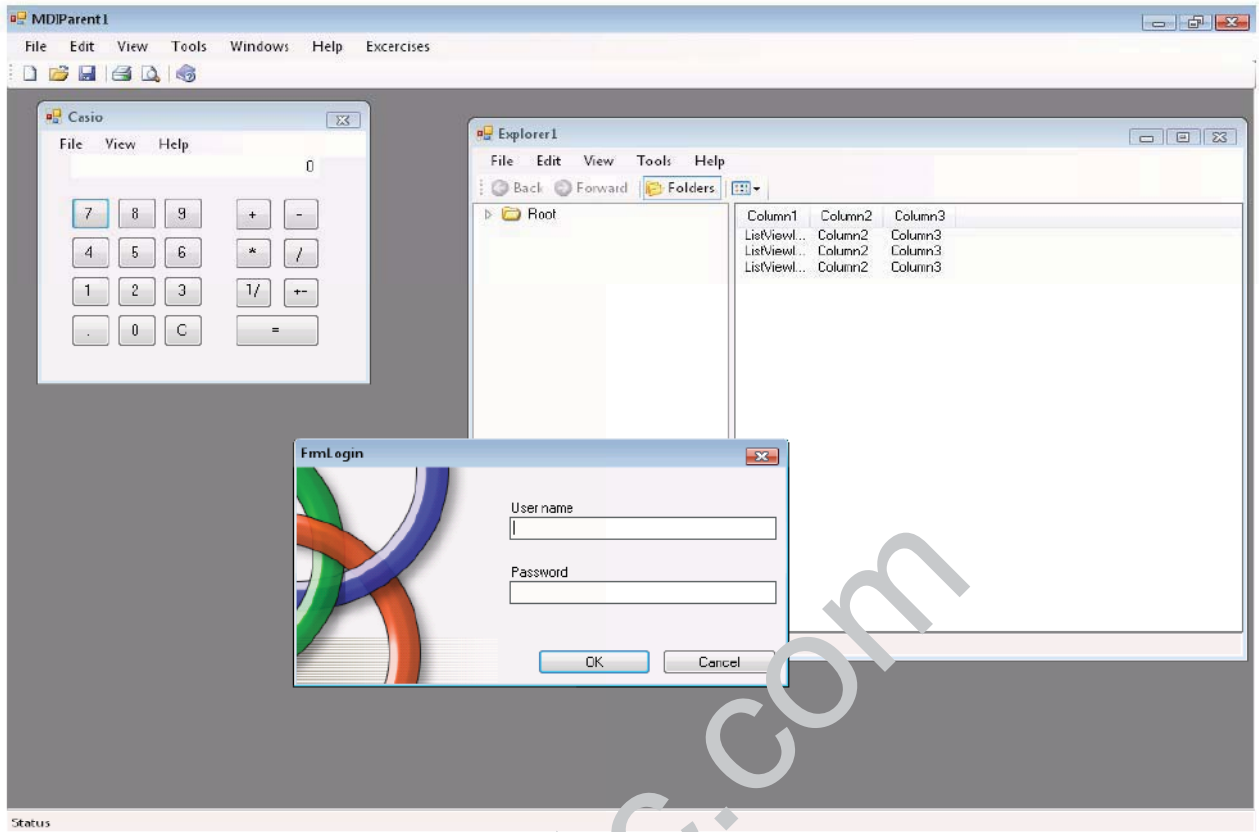
Source Code:

```
Private Sub SubMnuCasio_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
SubMnuCasio.Click
    Casio.StartPosition = FormStartPosition.CenterParent
    Casio.MdiParent = Me
    Casio.Show()
End Sub
```

```
Private Sub SubMnuLogin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
SubMnuLogin.Click
    FrmLogin.StartPosition = FormStartPosition.CenterParent
    FrmLogin.MdiParent = Me
    FrmLogin.Show()
End Sub
```

```
Private Sub FrmSubExplor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
FrmSubExplor.Click
    Explorer1.StartPosition = FormStartPosition.CenterParent
    Explorer1.MdiParent = Me
    Explorer1.Show()
End Sub
```

លទ្ធផល:

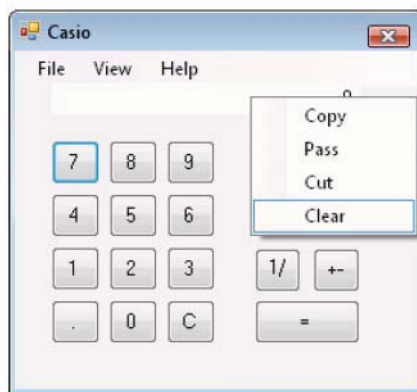


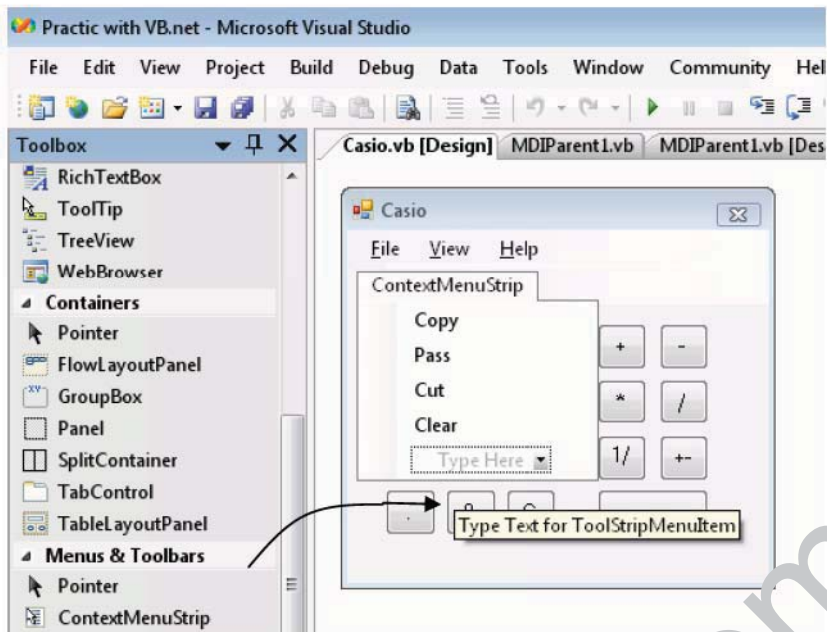
IX. របៀបបង្កើត Popup Menu នៅលើ Object

Popup Menu មានន័យថា នៅពេលដែលលោកអ្នកចុច Mouse ខាងស្តាំនៅលើ Object

ណាមួយអោយលេចចេញនូវទម្រង់ Menu ចេញមក។ ដើម្បីបង្កើតវាបានយើងត្រូវប្រើ Control មួយឈ្មោះថា ContextMenuStrip មកតភ្ជាប់លើ Form Object ហើយអោយតំលៃនៃ ContextMenuStrip Property នៃ Object ណាដែលយើងចង់បង្កើត Popup Menu មានតំលៃស្មើអោយឈ្មោះ ContextMenuStrip ដែលលោកអ្នកបានយកមកគូរ។

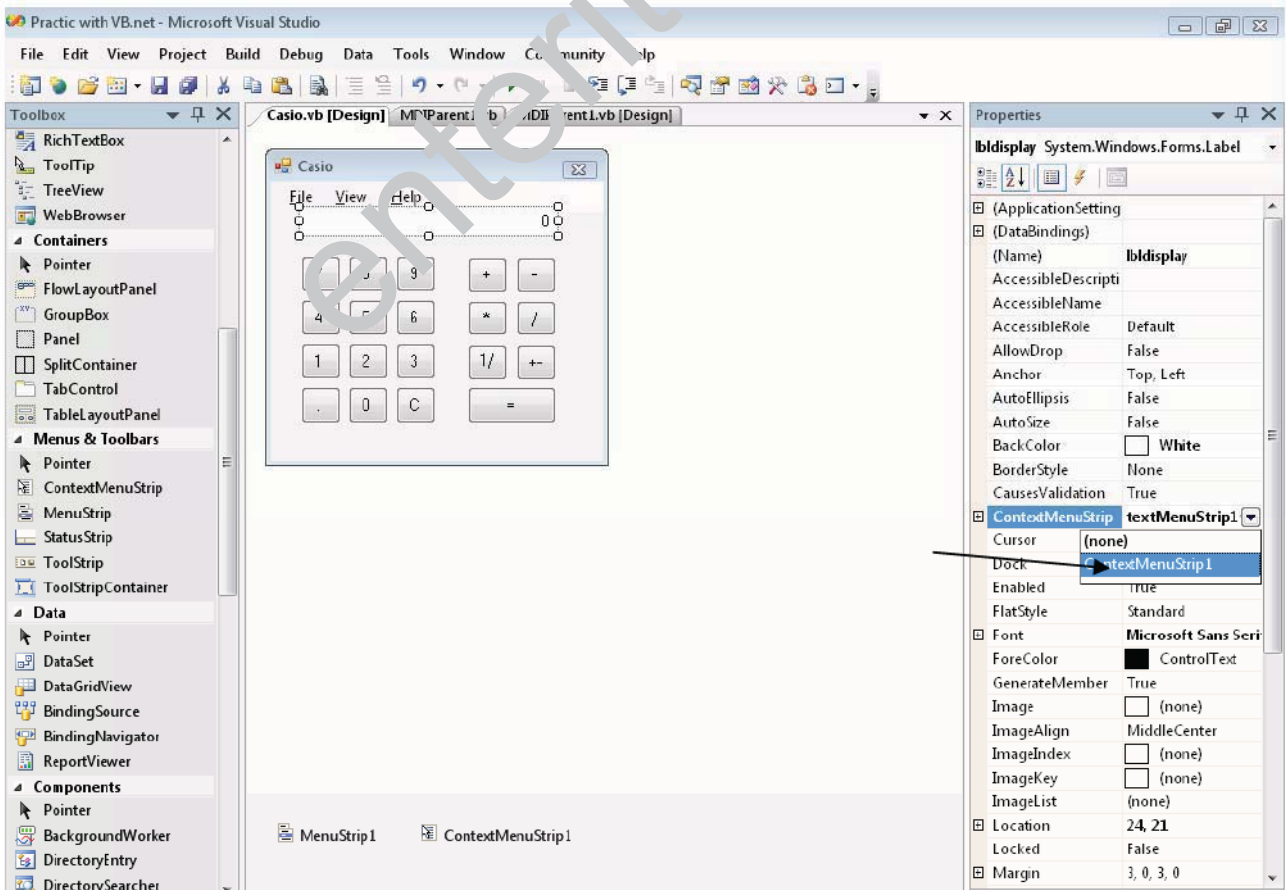
ឧទាហរណ៍: យើងបង្កើត Popup Menu មួយនៅលើtxtdisplay ដែលមានក្នុង FormCaculate ដូចរូប:





- បន្ទាប់មកសូមបង្កើត Item នីមួយៗដូចគ្នាទៅនឹង ការបង្កើត Menu ដែរ
- បន្ទាប់មកចុច Mouse ខាងស្តាំពិលី txtDisplay ដើម្បីកែលម្អនៃ ContextStrip Property របស់

txtDisplay ស្មើឈ្មោះ ContextMenuStrip ដែលយកមកប្រើ



-បន្ទាប់មកសូមដាក់ឈ្មោះអោយ Item នីមួយៗរបស់ ContextMenuStrip ដូចរូបខាងក្រោម:

Object	ដាក់ឈ្មោះអោយ ContextMenuItem
Copy	Cmscopy
Pass	Cmsspass
Cut	Cmscut
Clear	CmsClear

បន្ទាប់មកយើងសរសេរកូដទៅនឹង Object នីមួយៗដូចគ្នាទៅនឹង SubMenu Copy, Pass, Cut , Clear ។

enteritc.com

បេឡេនី ៣

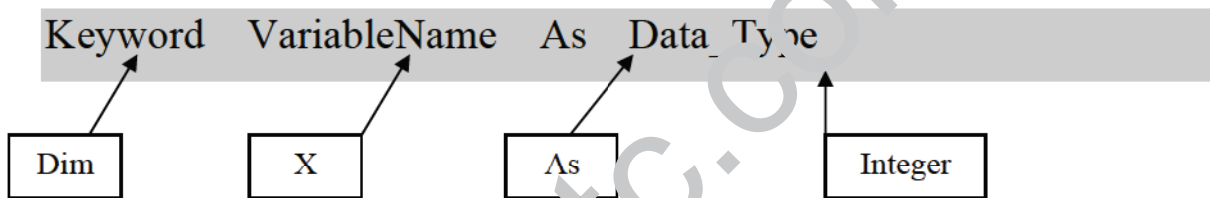
ការប្រើប្រាស់អថេរ

Using Variable to Store Information

I. អថេរ (Variable)

គ្រប់ភាសាសំរាប់សរសេរ Program ទាំងអស់មិនថាចំពោះភាសា VB.Net ទេតែងតែមាន Variable ដែលជាអាថ៌កំបាំងសំរាប់ទទួលយកតម្លៃ ហើយរាល់ Variable ទាំងអស់ត្រូវប្រើសម្រាប់រក្សាទុកតម្លៃជាបណ្តោះអាសន្ននៅពេលដែលកម្មវិធីកំពុងដំណើរការ ។

Syntax: 1



Syntax: 2

```
Dim Variable_Name1 Variable_Name2,..., As Data_Type
```

Syntax: 3

```
Dim Variable_Name As Data_Type, Variable_Name As Data_Type, Variable_Name2 As Data_Type,...
```



-Keyword ជា Statement ប្រើសម្រាប់កំណត់ របៀបប្រើប្រាស់របស់អញ្ញាតនោះ ។ វាមានដូចជា

- Private
- Public
- Dim
- Static

-As ជា Statement មួយដែលប្រើសំរាប់ កំណត់ប្រភេទ Data_Type ទៅអោយ អថេរ ។

-Data_Type ជាប្រភេទទិន្នន័យ ដែលអញ្ញាតរក្សាទុកបាន ។

Ex:

```
Dim x as double
Private LastName as String
Public k as Integer
Static I as Byte
```

ចំណាំ: គ្រប់ Variable ដែលប្រកាសនៅក្នុង General Declaration គឺជា Static variable

* **General Declaration** : ជាកន្លែងមួយសំរាប់ប្រកាស Variable ទិតនៅលើបំផុតនៃ Form Module គេអាចប្រើ Keyword Dim, Private, Public.

* **Static and Dim** (ក្នុង Procedure)

- + **Static variable** : អាចរក្សាតំលៃរបស់ Application ត្រូវបានបញ្ចប់ ។
- + **Dim variable** : អាចរក្សាតំលៃរបស់ Procedure ត្រូវបានបញ្ចប់ ។

តារាង Data Type

Data type	Size	Range	Sample usage
Short	16-bit	-32,768 through 32,767	Dim Birds As Short Birds = 12500
UShort	16-bit	0 through 65,535	
Integer	32-bit	-2,147,483,648 through 2,147,483,647	Dim Insects As Integer Insects = 37500000
UInteger	32-bit	0 through 4,294,967,295	Dim Joys As UInteger Joys = 3000000000
Long	64-bit	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Dim WorldPop As Long WorldPop = 4800000004

Data type	Size	Range	Sample usage
ULong	64-bit	0 through 18,446,744,073,709,551,615	Dim Stars As ULong Stars = 18000000000000000000
Single	32-bit floating point	-3.4028235E38 through 3.4028235E38	Dim Price As Single Price = 899.99
Double	64-bit floating point	-1.79769313486231E308 through 1.79769313486231E308	Dim Pi As Double Pi = 3.1415926535
Decimal	128-bit	values up to +/-79,228 x 1024	Dim Debt As Decimal Debt = 7600300.50
Byte	8-bit	0 through 255 (no negative numbers)	Dim RetKey As Byte RetKey = 13
SByte	8-bit	-128 through 127	Dim NegVal As SByte NegVal = -20
Char	16-bit	Any Unicode symbol in the range 0–(1,111,111)	Dim UnicodeChar As Char UnicodeChar = "Ä"
String	Usually 16-bits per character	0 to approximately 2 billion 16-bit Unicode characters	Dim Dog As String Dog = "pointer"
Boolean	16-bit	True or False (during conversions, 0 is converted to False, other values to True)	Dim Flag as Boolean Flag = True
Date	64-bit	January 1, 0001 through December 31, 9999	Dim Birthday as Date Birthday = #3/1/1963#
Object	32-bit	Any type can be stored in a variable of type Object	Dim MyApp As Object MyApp = CreateObject ("Word.Application")

◆ **Long**

ប្រើសំរាប់រក្សាទិន្នន័យជាជំនួសគត់ (ទំហំរបស់វាធំជាង Integer) ដែលមានបញ្ជាក់ទំហំវានៅតារាង Data Type ខាងលើ
ឧទាហរណ៍:

```

Source Code:
Private Sub Form_Load(...)
    Dim i AS Long
    i=999.7
    MsgBox i    '(i =1000)
End Sub
    
```

◆ **Single**

ប្រើសំរាប់រក្សាទិន្នន័យជាជំនួសទសភាគដែលមានបញ្ជាក់ទំហំ របស់វានៅក្នុងតារាង Data Type ខាងលើ ។

ឧទាហរណ៍:

```

Source Code:
Private Sub Form_Load(...)
    Dim i AS Single
    i=999.34
    MsgBox i '(i=999.34)
End Sub

```

◆ **Double**

ប្រើសំរាប់រក្សាទិន្នន័យជាមន្តសមភាគ (ទំហំធំជាង Single) ដែលមានបញ្ជាក់ទំហំ របស់វានៅ ក្នុងតារាង Data Type ខាងលើ ។

ឧទាហរណ៍:

```

Source Code:
Private Sub Form_Load(...)
    Dim i AS Double
    i=9899.398
    MsgBox i
End Sub

```

◆ **Currency**

ប្រើសំរាប់រក្សាទិន្នន័យប្រភេទជារូបិយវត្ថុ ។

ឧទាហរណ៍:

```

Source Code:
Private Sub Form_Load(...)
    Dim i AS Currency
    i=i
    MsgBox i
End Sub

```

◆ **String**

ប្រើសំរាប់រក្សាទិន្នន័យជាតួអក្សរ ឬ លេខ(មិនអាចគណនាបាន) ។ ចំពោះអថេរមួយបន្ទាប់ពីលោកអ្នកធ្វើការប្រកាសជ String គឺវាអាចរក្សាទិន្នន័យអក្សររហូតដល់ទំហំជិត ២ GB ។ **ចំណាំ:** រាល់តំលៃផ្ទេរអោយអថេរ ត្រូវតែនៅក្នុងសញ្ញា

“ ” **ឧទាហរណ៍:**

```

Source Code:
Private Sub Form_Load(...)
    Dim str,Tel AS String
    Tel="012 455 888"

```

```

str=" Welcome to Cnit Center "
MsgBox str & Tel
End Sub

```

◆ **Boolean**

ប្រើសំរាប់រក្សាទិន្នន័យជាជំរើស ពិត ឬ មិនពិត ។ **ចំណាំ:** True មានតំលៃស្មើ 1 ហើយ False មានតំលៃស្មើ 0

ឧទាហរណ៍:

☞ Source Code:

```

Private Sub Form_Load(...)
Dim i AS Boolean
i=true ' Or i=1
If i =true then
MsgBox(i)
Else
MsgBox (i)
End if
End Sub

```

◆ **Date**

ប្រើសំរាប់រក្សាទិន្នន័យជាថ្ងៃខែឆ្នាំ ព្រមទាំងម៉ោង ។

ឧទាហរណ៍:

☞ Source Code:

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
Dim d1 As Date
d1 = #5/12/2008#
MsgBox(d1.ToString("D"), ' => Monday, May,12,2008
d1 = d1.AddDays(5)
MsgBox(d1.ToString("d")) '=> 17-May-08
d1 = d1.AddMonths(3)
MsgBox(d1.ToString(d1.ToString("MMM/dd/yyyy"))) '=> Agu-15-2008
End Sub

```

◆ **Object**

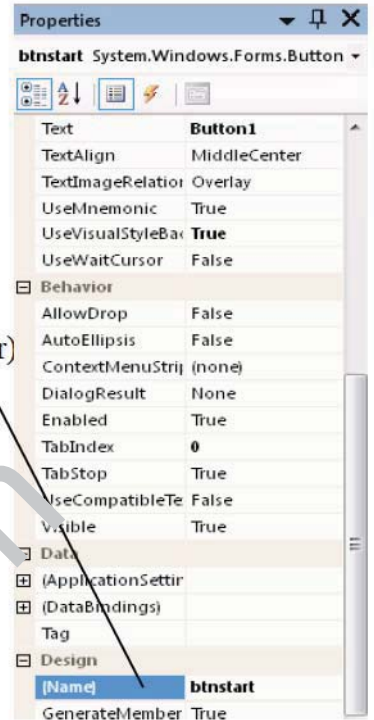
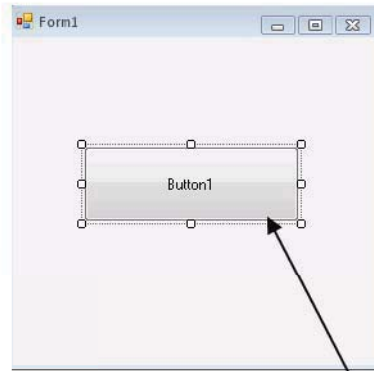
ប្រើសំរាប់ទុកទិន្នន័យប្រភេទជា Object ។

ឧទាហរណ៍:

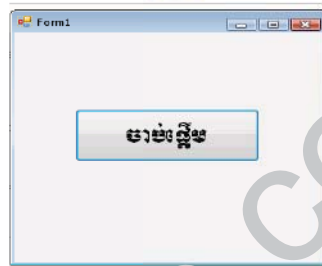


Source Code:

```
Private Sub Form_Load(...)
    Dim Btn AS Object
    Btn=Me.btnstart
    Btn.text="ចាប់ផ្តើម"
    Btn.Font=New Drawing.Font("Limon R1",30,Regular)
End Sub
```



លទ្ធផល



II. របៀបកំណត់ទិន្នន័យប្រភេទជា Number និង Date

a. កំណត់ទិន្នន័យប្រភេទជា Number

ដើម្បីកំណត់ទំរង់ទិន្នន័យទៅអោយ Number យើងត្រូវអនុវត្តតាមទំរង់ Format ដូចរូបខាងក្រោម :

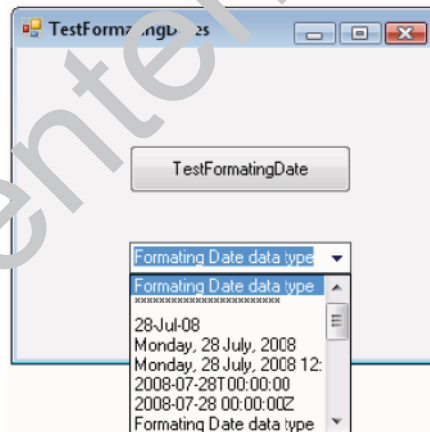
Format Character	អធិប្បាយ	ឧទាហរណ៍
C or c	Currency	12345.67.ToString("C") លទ្ធផល: \$12,345.67
E or e	Scientific	12345.67.ToString("E") លទ្ធផល: 1.234567E+004
F or f	Fixed-point format	12345.67.ToString("F") លទ្ធផល: 12345.67
G or g	General Format	លទ្ធផលអាចជា Fomat Character E or F
N or n	Number format	123456.67.ToString("N") លទ្ធផល: 12345.67
X or x	Hexadecimal format	250.ToString("X") លទ្ធផល: FA

enteritc.com

b. ការកំណត់ទំរង់លើទិន្នន័យប្រភេទថ្ងៃ Dates

Format Character	អតិប្បាយ	Format
D	Short date format	MM/dd/yyyy
D	Long date format	dddd, MMMM dd, yyyy
f	Long date followed By short time	dddd,MMMM dd, yyyy HH:mm
F	Long date followed By long time	dddd, MMMM dd, yyyy HH:mm:ss
g	(General) Short date followed by Short time	MM/dd/yyyy HH:mm
G	(General) Short Date followed by long time	MM/dd/yyyy HH:mm:ss
M or m	Month/day format	MMMM dd
R or r	RFC1123 patten	ddd, dd MMM yyyy HH:mm:ss GMT
S	Shortable date/time Format	Yyyv MM-dd HH:mm:ss
t	Short time format	HH:mm
T	Long time format	HH:mm:ss
u	Universal date/time	Yyyv-MM-dd HH:mm:ss
U	Universal short able date/time format	dddd,MMMM dd, yyyy HH:mm:ss
Y or y	Year month format	MMMM, yyyy

ឧទាហរណ៍:



Source Code:

```
Private Sub btnTest_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTest.Click
    Me.compl.Items.Add("Formatting Date data type")
    Me.compl.Items.Add("*****")
    Me.compl.Items.Add(Date.Today.ToString("d"))
    Me.compl.Items.Add(Date.Today.ToString("D"))
    Me.compl.Items.Add(Date.Today.ToString("f"))
    Me.compl.Items.Add(Date.Today.ToString("s"))
    Me.compl.Items.Add(Date.Today.ToString("u"))
End Sub
```

III. អប្បបរមាបំណែងពីប្រភេទទិន្នន័យមួយទៅប្រភេទទិន្នន័យមួយ

ចំពោះហិកាសរសេរ Program លាក់អ្នកតែងតែជួបនូវស្ថានភាពណាមួយដែលទាមទារអោយយើង

ធ្វើការបំណែងពីប្រភេទទិន្នន័យមួយទៅជាប្រភេទទិន្នន័យមួយផ្សេងទៀតដើម្បីយកទៅធ្វើការងារផ្សេងៗ ។ ភាសា

Visual Basic អនុញ្ញាតអោយយើងអ្នកប្រើនូវអនុគមន៍មួយចំនួនដូចតារាងខាងក្រោម:

អនុគមន៍	អនិច្ចាស
CBool	ប្រើសំរាប់បំណែងទិន្នន័យជា Boolean
CByte	ប្រើសំរាប់បំណែងទិន្នន័យជា Byte
CChar	ប្រើសំរាប់បំណែងទិន្នន័យជា Character
CDate	ប្រើសំរាប់បំណែងទិន្នន័យជា Date
CDbl	ប្រើសំរាប់បំណែងទិន្នន័យជា Double
CDec	ប្រើសំរាប់បំណែងទិន្នន័យជា Decimal
Cint	ប្រើសំរាប់បំណែងទិន្នន័យជា Integer(4-byte Integer, Int 32)
CLng	ប្រើសំរាប់បំណែងទិន្នន័យជា Long(8-byte Integer, Int 64)
CShort	ប្រើសំរាប់បំណែងទិន្នន័យជា Short(2-byte Integer, Int16)
Csng	ប្រើសំរាប់បំណែងទិន្នន័យជា Single
CStr	ប្រើសំរាប់បំណែងទិន្នន័យជា String
CObj	ប្រើសំរាប់បំណែងទិន្នន័យជា Object

IV. ការប្រើ User Defined Data Type

User Defined Data Type ជាអថេរមួយដែលអាចរក្សាទុកប្រភេទទិន្នន័យច្រើនផ្សេងៗគ្នា ។

ដើម្បីប្រកាសអថេរមួយដែលមានប្រភេទទិន្នន័យលើសពីមួយបាន ភាសា Visual Basic ទាមទារអោយអ្នកប្រើប្រាស់នូវ User-Defined Data Type ដែលមានទម្រង់ដូចខាងក្រោម:

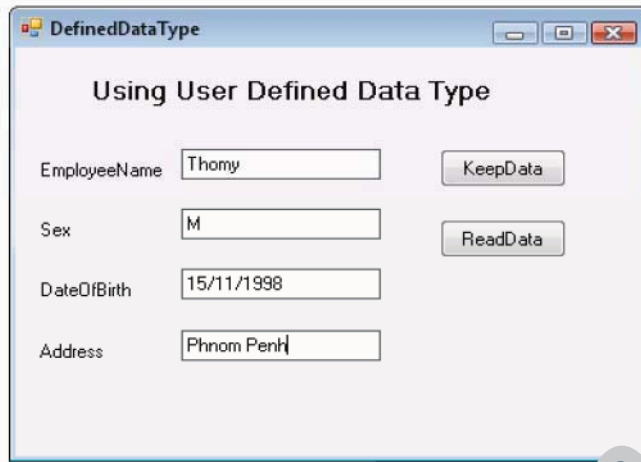
Syntax:

```

Structure structureName
    Dim Variable1 As VarType
    Dim Variable2 As VarType
    .....
    Dim Variablen As VarType
End Structure

```

ឧបមាធាតុយើងមាន Form ដូចខាងក្រោម:



Source Code:

```

Structure Employee
    Dim EmpName As String
    Dim sex As String
    Dim DateOfBith As Date
    Dim Address As String
End Structure

Dim EmpObj As Employee
.....
Sub clear()
    Me.txtaddress.Clear()
    Me.txtdob.Clear()
    Me.txtname.Clear()
    Me.txtsex.Clear()
End Sub
.....
Private Sub btnkeep_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnkeep.Click

    EmpObj.EmpName = Me.txtname.Text
    EmpObj.sex = Me.txtsex.Text
    EmpObj.DateOfBith = Me.txtdob.Text
    EmpObj.Address = Me.txtaddress.Text
    Call clear()
End Sub
.....
Private Sub btnread_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnread.Click
    Me.txtname.Text = EmpObj.EmpName
    Me.txtsex.Text = EmpObj.sex
    Me.txtdob.Text = EmpObj.DateOfBith
    Me.txtaddress.Text = EmpObj.Address
End Sub
.....

```

V. ការប្រើអថេរដែលមានតំលៃថេរ Constant Variable Types

នៅក្នុង VB អ្នក Programmer បានបង្កើត Variable ដែលតំលៃរបស់វាមិនអាចផ្លាស់ប្តូរបាននៅកំឡុងពេល Program កំពុងដំណើរការដែលគេហៅថា Constant Variable ។

ដើម្បីធ្វើការប្រកាសអថេរដែលមានតំលៃថេរ លោកអ្នកត្រូវប្រើនូវ Syntax ដូចខាងក្រោម:

```
Const constantName As Type=Value
```

ចំណាំ: ចំពោះអថេរអថេរទាំងឡាយណាដែលមានលក្ខណៈជា Constant មានតំលៃរបស់អថេរនោះមិនអាចធ្វើការផ្លាស់ប្តូរបានទេ

ឧទាហរណ៍:

```
Const Pi As Double=3.14159265358979
```

VI. កំណត់ទ្វេដំណើរការរបស់អថេរដែលមានតំលៃថេរ (Variable Scope & Life Time of Variable)

Variable Scope គឺជាដែនកំណត់នៃការប្រើប្រាស់ Variable ទៅតាមតំបន់នីមួយៗ ដែលធ្វើអោយ Variable មានឥទ្ធិពល ឬ គ្មានឥទ្ធិពល ចំពោះការប្រកាសតាមតំបន់ ។

a. Level of Variable

Level	Description
-Block Variable	: គឺជា Variable ដែលប្រកាសនៅក្នុង Block (Control Structure, Loop...) ជាមួយ Keyword Dim ឬ Static * អថេរនេះប្រើបានតែក្នុង Block ប៉ុណ្ណោះ ។
-Module Variable	: គឺជា Variable ដែលប្រកាសនៅក្នុង Module ជាមួយ Keyword Private , Public or Dim . * អថេរនេះប្រើបានតែនៅក្នុង Module(Form Module)
-Local Variable	: គឺជា Variable ដែលប្រកាសនៅក្នុង Sub Procedure ជាមួយ Keyword Private or Dim .

* អថេរនេះប្រើបានទាល់តែបង្កើត **Object** ជាមុនសិនទើបយើងអាច **Access** ទៅកាន់អថេរដែលបានប្រកាសនៅក្នុង **Class** នោះ ។

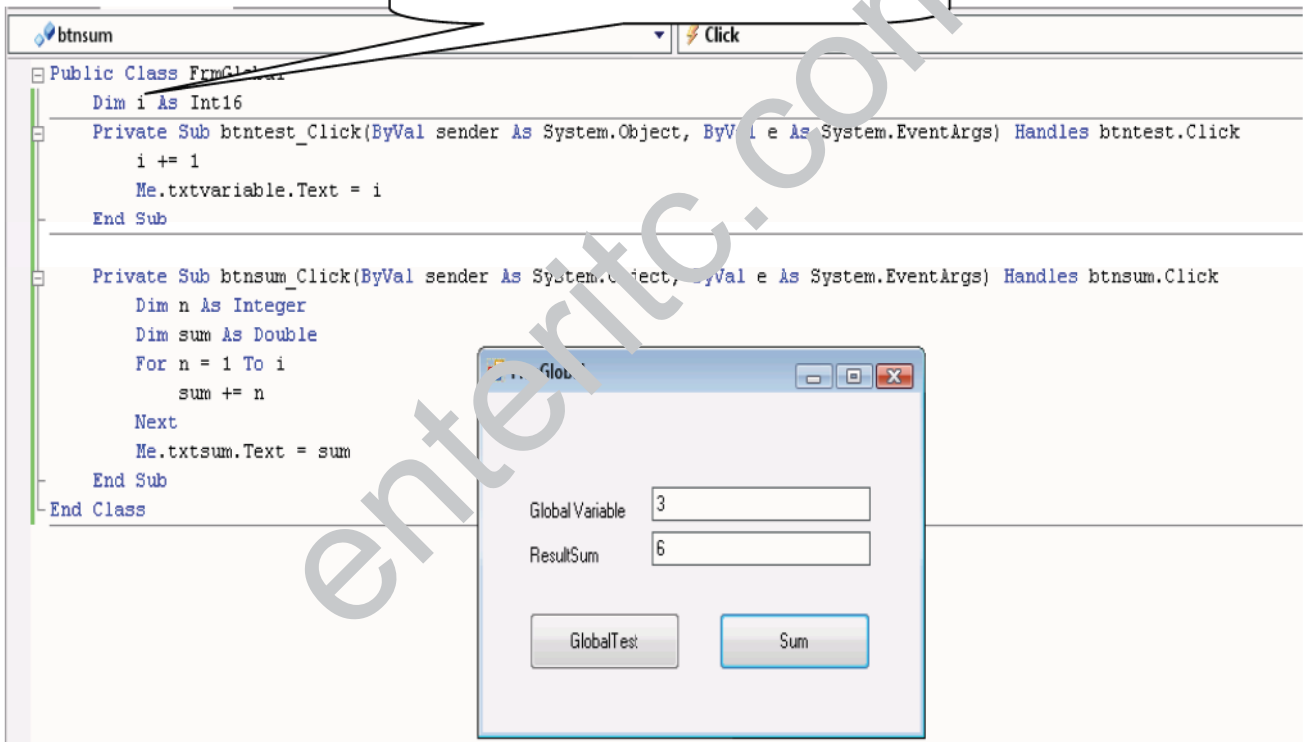
-Global Variable : គឺជា **Variable** ដែលប្រកាសនៅក្នុង **Standard Moudule** ជាមួយ **Keyword Public**

* អថេរនេះប្រើបានគ្រប់កន្លែងក្នុង **Project**

☞ 1-កំរិត **Global** ធ្វើការប្រកាសដោយចាប់ផ្តើមដោយ **Keyword Public** ឬ **Dim** នៅក្នុងផ្នែក **General Delclaration** ហើយរាល់អថេរទាំងឡាយណាក៏ដោយដែល កំរិតដំណើរការរបស់វាជា **Global** តំលៃរបស់វាត្រូវបាន **Clear** ចេញពី **Memory** ដោយគ្រាន់តែ បិទ **Application** ហើយអថេរនោះអាចប្រើបានគ្រប់ **Procedure** ទាំងអស់ ដែលមាននៅក្នុង **Form Module** ។

ឧទាហរណ៍:

i ជា **Variable** ប្រភេទ **Global Variable**

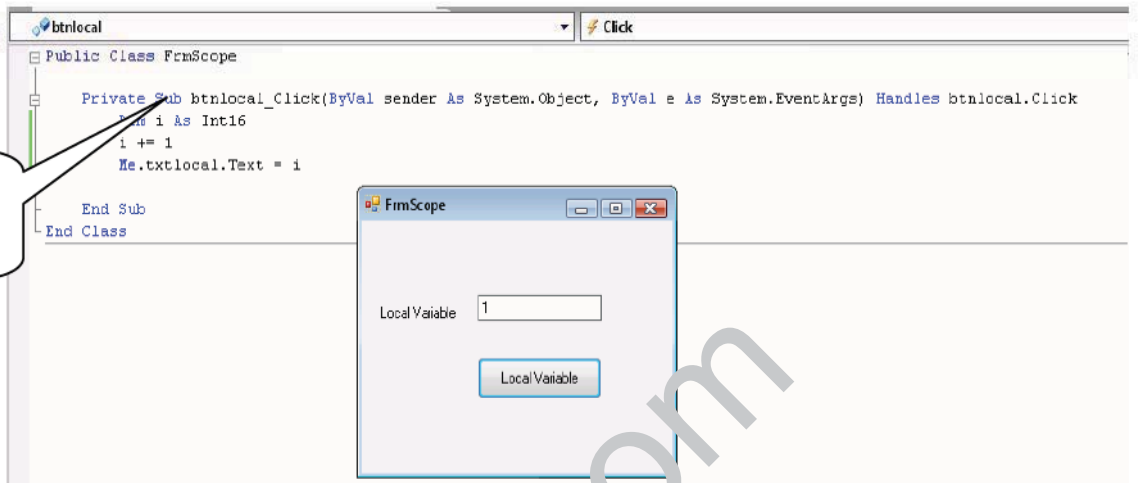


☞ 2- កំរិត **Local Variable**

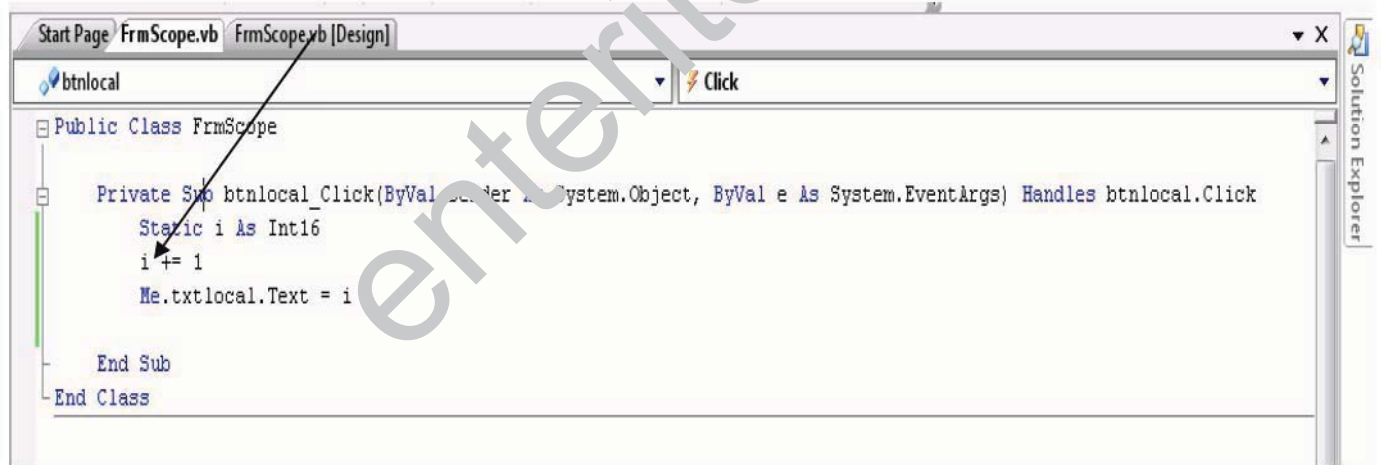
លោកអ្នកអាចធ្វើការប្រកាសអថេរដែលមានលក្ខណៈ **Local** បានដោយធ្វើការប្រកាសវានៅក្នុង **Sub Procedure** ដោយចាប់ផ្តើមដោយ **Keyword Private** ឬ **Dim** ។ ចំពោះ **Global Variable** តំលៃរបស់វាត្រូវបាន **Clear** ចេញពី

Memory នៅពេលដែលដំណើរការ Procedure ត្រូវបានបញ្ចប់ ហើយ Variable នេះអាចប្រើបានតែនៅក្នុង Sub Procedure ដែលបានប្រកាសវាតែប៉ុណ្ណោះ ។

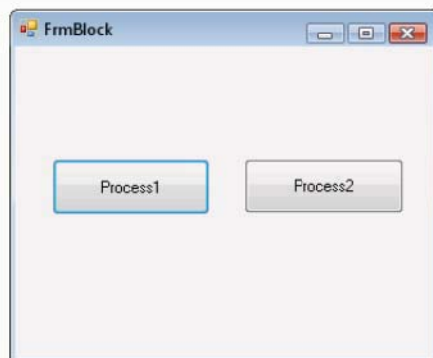
ឧទាហរណ៍:

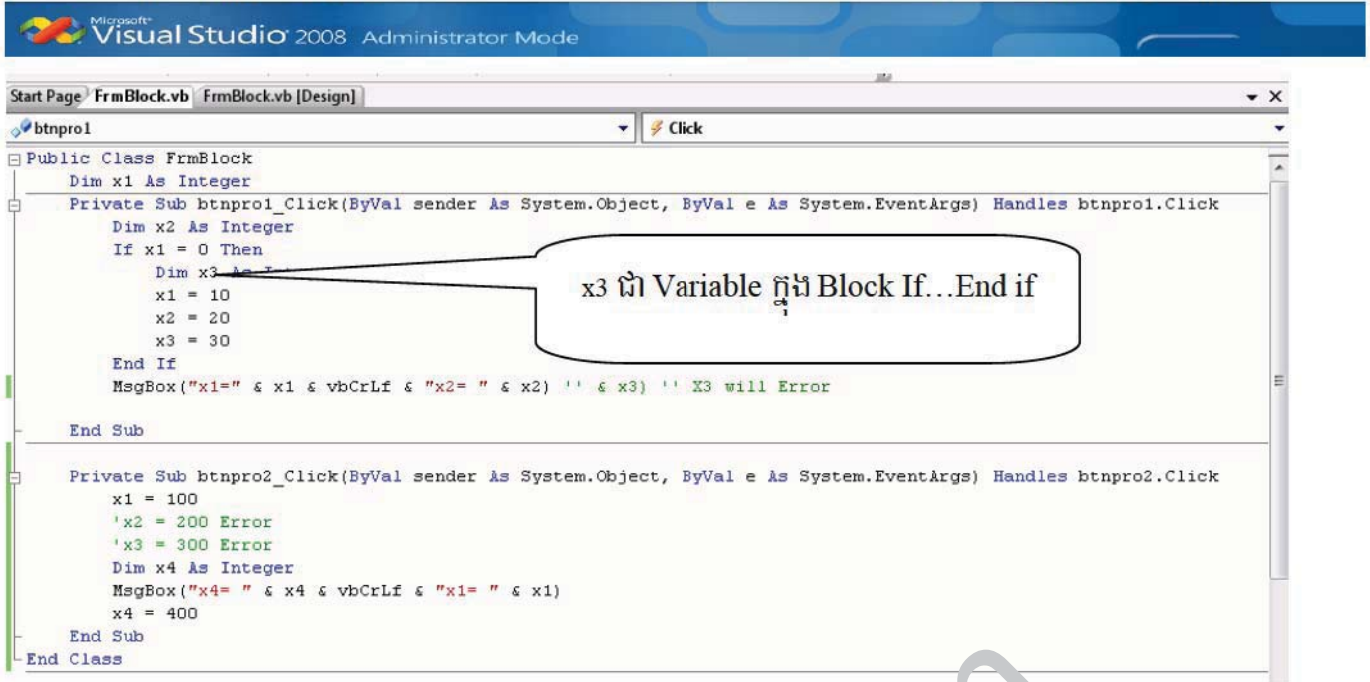


ចំណាំ: ចំពោះ Local Variable លោកអ្នកអោយតំលៃរបស់វារក្សាទុកនៅក្នុង Memory រហូតដល់បិទ Application លុះត្រាតែលោកអ្នកធ្វើការប្រកាសវា ដោយប្រើ Keyword Static.



3- Block Variable: គឺជា Variable ប្រកាសក្នុង Block ជាមួយ Keyword Dim





4- Module Variable

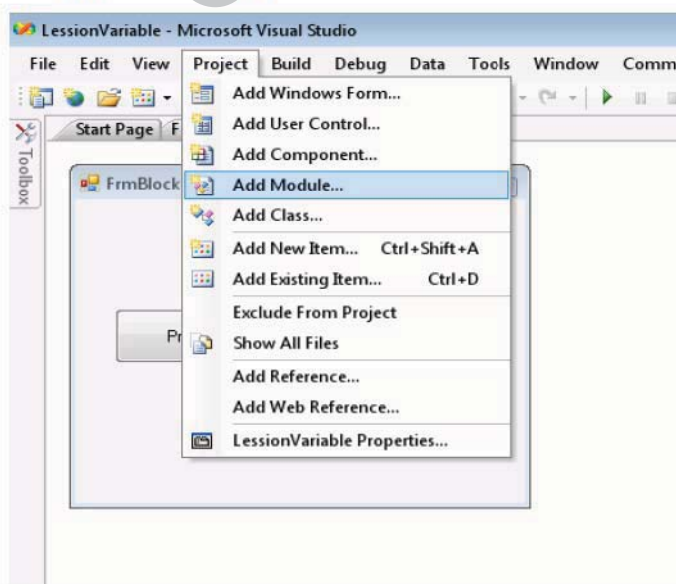
មុននឹងធ្វើការសិក្សាពីកំរិតដំណើរការរបស់អថេរលោកអ្នកត្រូវដឹងពី Module ជាមុនសិន ។ នៅក្នុងភាសា Visual

Basic Module ត្រូវបានបែងចែកជាបីគឺ:

- 1- Standard Module
- 2- Form Module
- 3- Class Module

4-1 Standard Module

ជា Object មួយដែលអនុញ្ញាតអោយលោកអ្នកធ្វើការបង្កើតនូវ Sub Program (Procedures) ឬ ប្រកាសអថេរដែលអថេរនោះអាចយកទៅប្រើបានគ្រប់ទីកន្លែង។ នៅ Form Module ទាំងអស់ដោយប្រើ Keyword Public ។ ប៉ុន្តែប្រសិនបើអថេរទាំងនេះត្រូវបានប្រកាសដោយប្រើ Keyword Private វិញមានន័យថា អថេរទាំងនោះអាចប្រើបានតែក្នុង Standard Module ប៉ុណ្ណោះ។ ដើម្បីបង្កើត Standard Module



4-2 Form Module

ជា Object មួយប្រើសំរាប់បង្កើត Sub Program ឬ ប្រកាស Variable តែវាមាន Interface សំរាប់អោយយើងប្រើប្រាស់ និង Event សំរាប់ធ្វើការបញ្ជាទៅលើ Object ដែលមាននៅលើ Form ។

4-3 Class Module

ជាកន្លែងមួយ សំរាប់អោយយើងបង្កើតនូវ Object, Properties, Method, Event ដោយខ្លួនឯង ។

VI. និមិត្តសញ្ញាប្រមាណវិទ្យា:

ការប្រើប្រាស់សញ្ញាប្រមាណវិទ្យានៅក្នុង VB មានការខុសប្លែកពីប្រមាណវិទ្យានៅក្នុងគណិតវិទ្យាខ្លះដែរ ។ និមិត្តសញ្ញាដែលគេប្រើក្នុង VB រួមមាន:

Operators	VB Expression	Arithmetic
^	A ^ B	A^B
*, /	A * B, A / B	$A \times B, \frac{A}{B}$
\	A \ B (Integer)	A \ B (យកលេខគត់)
+, -	A + B, A - B	A + B, A - B
Mod	A Mod B	លទ្ធផលជាសំណល់

ឧ:

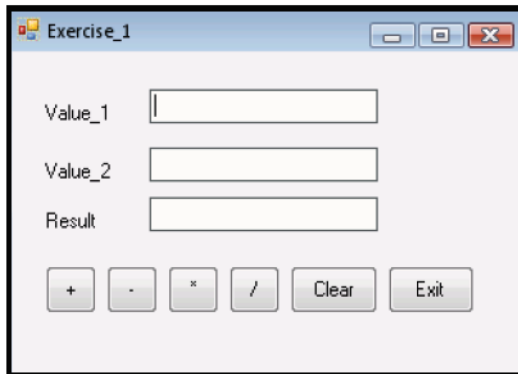
- $2^3 (2^3) = 8$
- $3 * 4 (3 \times 4) = 12$
- $5 / 2 (\frac{5}{2}) = 2.5$
- $5 \setminus 2 = 2$
- $2 + 5 = 7$
- $2 - 5 = -3$
- $5 \text{ Mod } 2 = 1$

បេឡេនី ៤

Using Variable in Programming

V. ការសរសេរកម្មវិធី

១. ចូរសរសេរកម្មវិធីសម្រាប់គណនាផលបូក, ដក, គុណ, ចែក នៃពីរចំនួន



Object	Properties Name	Properties Setting
Label1	Text	Value_1
Label2	Text	Value_2
TextBox1	Name Text	Txtvalue1 (Empty)
TextBox2	Name Text	Txtvalue2 (Empty)
Button1	Name Text	Btnsum +
Button2	Name Text	Btnsub -
Button3	Name Text	Btnmulti *
Button4	Name Text	Btndivi /
Button5	Name Text	Btnclear Clear
Button6	Name Text	BtnExit Exit

```

Private Sub btnsum_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsum.Click
    Dim A As Double, B As Double, S As Double
    A = Me.txtval1.Text
    B = Me.txtval2.Text
    S = A + B
    Me.txtresult.Text = S
End Sub
    
```



```
Private Sub btnsub_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsub.Click
    Dim A As Double, B As Double, S As Double
    A = Me.txtval1.Text
    B = Me.txtval2.Text
    S = A - B
    Me.txtresult.Text = S
End Sub
```

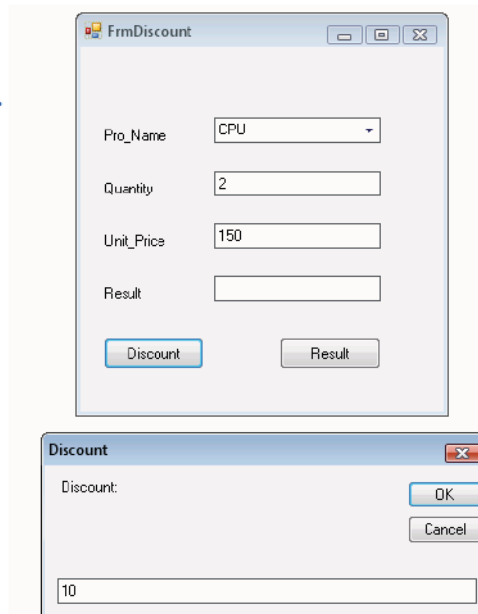
```
Private Sub btnmulti_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnmulti.Click
    Dim A As Double, B As Double, S As Double
    A = Me.txtval1.Text
    B = Me.txtval2.Text
    S = A * B
    Me.txtresult.Text = S
End Sub
```

```
Private Sub btndivi_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndivi.Click
    Dim A As Double, B As Double, S As Double
    A = Me.txtval1.Text
    B = Me.txtval2.Text
    S = A / B
    Me.txtresult.Text = S
End Sub
```

```
Private Sub btnclear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnclear.Click
    Me.txtval1.Clear()
    Me.txtval2.Text = ""
    Me.txtresult.Clear()
    Me.txtval1.Focus()
End Sub
```

```
Private Sub btnexit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnexit.Click
    'End
    Me.Close()
End Sub
```

២. គេមាន Application ដូចខាងក្រោម:



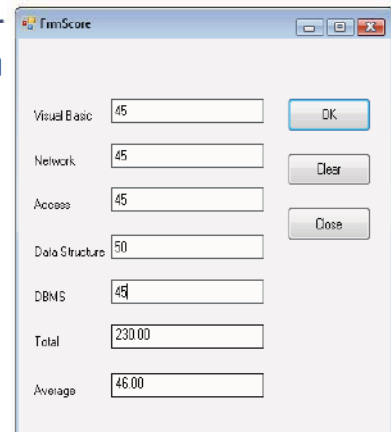
-ការសរសេរកូដ

```

Dim Discount, Price As Double
.....
Private Sub btndiscount_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndiscount.Click
Discount = InputBox("Discount:", "Discount")
End Sub
.....
Private Sub btnresult_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnresult.Click
Dim Qty As Integer, UnitPrice As Double
Qty = Me.txtQuantity.Text
UnitPrice = Me.txtprice.Text
Me.txtresult.Text = Format((Qty * UnitPrice) - Discount, "##,##0.00$")
End Sub
.....
Private Sub FrmDiscount_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
Me.ComPro.Items.Add("Hard Disk")
Me.ComPro.Items.Add("CPU")
Me.ComPro.Items.Add("RAM")
Me.ComPro.Items.Add("Monitor")
Me.ComPro.Items.Add("CD-ROM")
Me.ComPro.Items.Add("Printer")
Me.ComPro.Items.Add("Scanner")
End Sub

```

៣. ចូរសរសេរកម្មវិធីសម្រាប់គណនាពិន្ទុសរុប និង ចម្រុះសាមី នៃ ៥ មុខវិជ្ជា



-ការសរសេរកូដ

```

Private Sub btnok_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click
Dim vb, Access, Network, DBMS, DS As Double
vb = Me.txtvb.Text
Access = Me.txtaccess.Text
Network = Me.txtnet.Text
DBMS = Me.txtdbms.Text
DS = Me.txtdata.Text

Dim Total As Double
Total = vb + Access + Network + DBMS + DS

Dim Average As Double
Average = Total / 5

```



```
Me.lbltotal.Text = Format(Total, "##,##0.00")
Me.lblavg.Text = Format(Average, "##,##0.00")
```

End Sub

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

```
Me.txtvb.Clear()
Me.txtnet.Clear()
Me.txtaccess.Clear()
Me.txtdata.Clear()
Me.txtdbms.Clear()
Me.lblavg.Text = ""
Me.lbltotal.Text = ""
Me.txtvb.Focus()
```

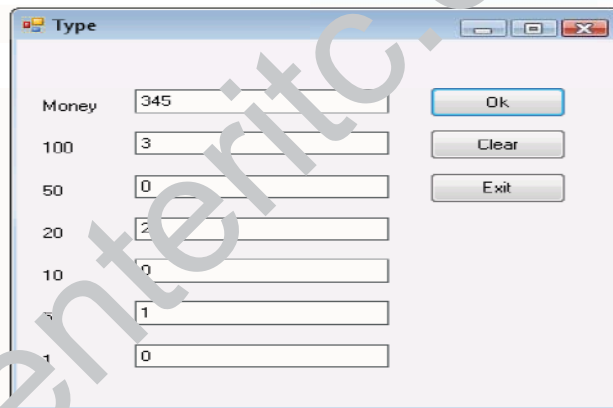
End Sub

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

```
Me.Close()
```

End Sub

៤. ចូរសរសេរកម្មវិធីសម្រាប់រកចំនួនក្រដាសប្រាក់ ១០០, ៥០, ២០, ១០, ៥ និង ១ ដុល្លារ ដែលមានទំរង់ដូចខាងក្រោម:



-ការសរសេរកូដ

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

```
Me.txtmoney.Clear()
Me.txt100.Clear()
Me.txt50.Clear()
Me.txt20.Clear()
Me.txt10.Clear()
Me.txt5.Clear()
Me.txt1.Clear()
Me.txtmoney.Focus()
```

End Sub

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

```
Dim Money As Integer
Money = Me.txtmoney.Text
```



```

Me.txt100.Text = (Money \ 100).ToString
Me.txt50.Text = (Money Mod 100) \ 50
Me.txt20.Text = ((Money Mod 100) Mod 50) \ 20
Me.txt10.Text = (((Money Mod 100) Mod 50) Mod 20) \ 10
Me.txt5.Text = ((((Money Mod 100) Mod 50) Mod 20) Mod 10) \ 5
Me.txt1.Text = ((((((Money Mod 100) Mod 50) Mod 20) Mod 10) Mod 5) \ 1
End Sub

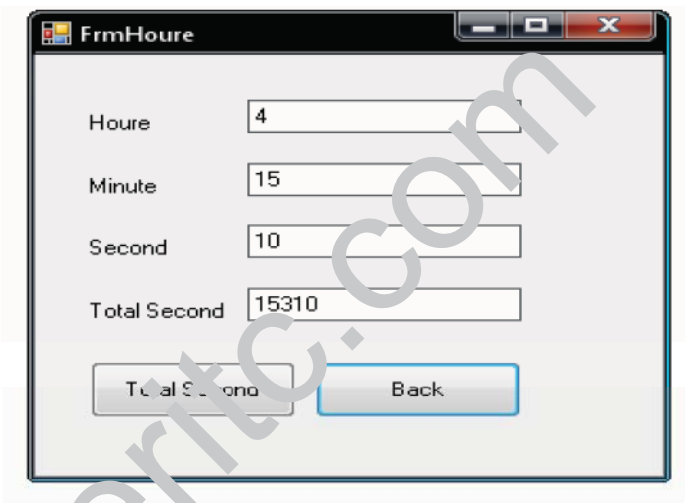
```

```

.....
Private Sub btnexit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnexit.Click
    End
End Sub
.....

```

៥-សរសេរកម្មវិធីរក Second ហើ ប៉ុន្តែ Back រកជា Hour, Minute និង Second ដូចរូបខាងក្រោម:



-ការសរសេរ Code:

```

.....
Private Sub btntotalsecond_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btntotalsecond.Click

    Me.txttotalsecond.Text = (Val(Me.txthour.Text) * 3600) + (Val(Me.txtminute.Text) * 60) +
(Val(Me.txtsecond.Text))
    Me.txthour.Clear()
    Me.txtminute.Clear()
    Me.txtsecond.Clear()

```

End Sub

```

.....
Private Sub btnback_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnback.Click

    Me.txthour.Text = Val(Me.txttotalsecond.Text) \ 3600
    Me.txtminute.Text = (Val(Me.txttotalsecond.Text) - Val(Me.txthour.Text) * 3600) \ 60
    Me.txtsecond.Text = Val(Me.txttotalsecond.Text) - ((Val(Me.txthour.Text) * 3600) +
(Val(Me.txtminute.Text) * 60))

```

End Sub

មេរៀនទី ៥

Control Structure

Control Statement ប្រើសំរាប់អនុញ្ញាតឱ្យយើង Control លើការអនុវត្តន៍ Code ក្នុង Programme (បញ្ហាទិសដៅនៃការអនុវត្តន៍ Code) ។

Control Statement មាន

1. Selection Statement
2. Looping Statement
3. Jumping Statement

I. Selection Statement

សំរាប់បញ្ហាទិសដៅនៃការអនុវត្តដោយការជ្រើសរើសទៅតាមខ្លឹមសាររបស់ប្រើប្រាស់។

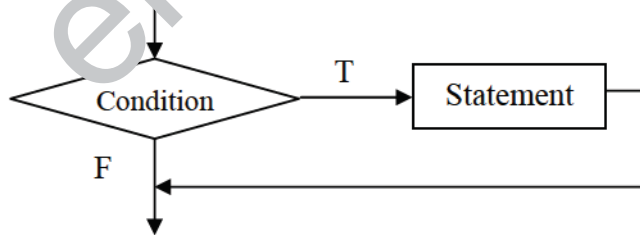
- A. If...Then
- B. Select Case...

A. If...Then

Syntax:

1. If Condition Then Statement
2. If Condition Then Statement
End if

Flow Chart:



Ex:

```

If(A>=B) Then
  MsgBox("Max of A and B is A")
End if

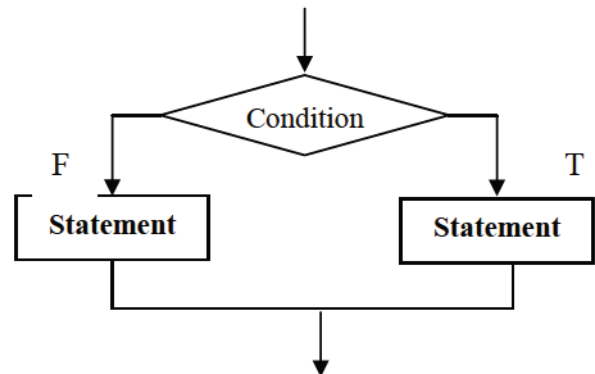
```

3. If Condition Then

```

Statements
Else
  Statements
End if

```





Ex:

```

If(A>=B) Then
  MsgBox("Max of A and B is A")
Else
  MsgBox("Max of A and B is B")
End if

```

4. Multi-Condition:

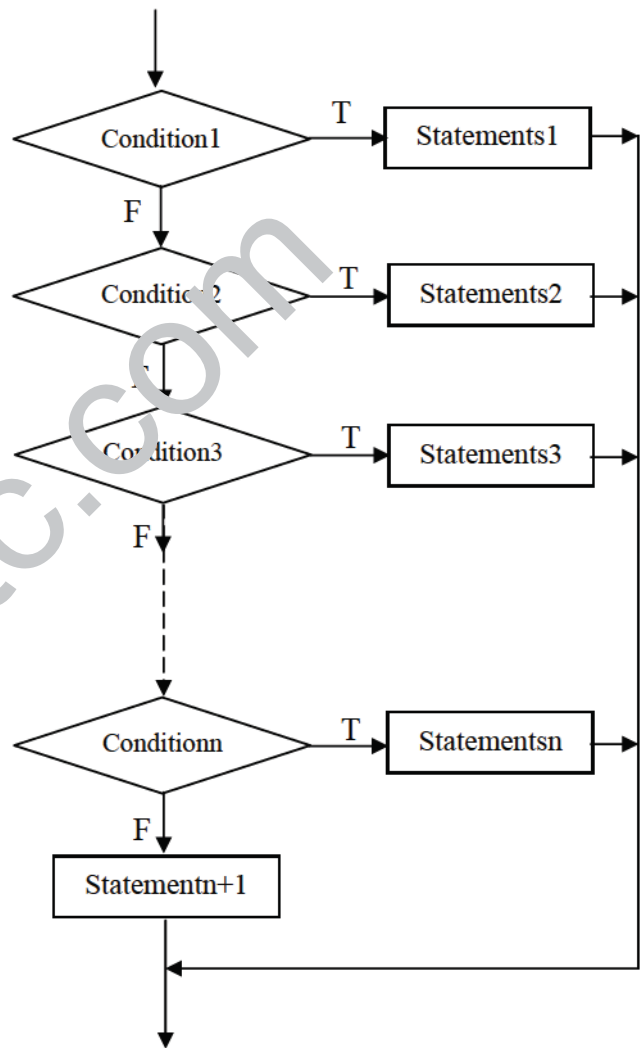
Syntax:

```

If Condition 1 Then
  [Statement 1]
Elseif Condition 2 Then
  [Statement 2]
Elseif Condition3 Then
  [Statement 3]
  :
  :
Elseif Condition n Then
  [Statement n]
Else
  [Statement n+1]
End if

```

Flow Chart:



Ex:

```

If (Avg>=90) and (Avg<=100) Then
  TxtM.Text= "Excelent"
Elseif(Avg>=80) Then
  TxtM.Text="Very Good"
Elseif(Avg>=70) Then
  TxtM.Text="Good"
Elseif(Avg>=60) Then
  TxtM.Text="Fair"
Elseif(Avg>=50) Then
  TxtM.Text="Pass"
Else
  TxtM.Text="Fail"
End If

```

B. Slect Case Statement:

Syntax:

```

Select Case Expression
  Case Exp1
    [Statements 1]
  Case Exp2
    [Statements 2]
  :
  :
  Case Else
    [Statements n]
End Select

```

Ex:1

Select Case Num

```

Case 1
    TxtR.Text="One"
Case 2
    TxtR.Text="Two"
Case 3
    TxtR.Text="Three"
Case 4
    TxtR.Text="Four"
Case 5
    TxtR.Text="Five"
Case 6
    TxtR.Text="Six"
Case Else
    TxtR.Text="Unknwon"

```

End Select

Ex:2

Select Case Age.Text

```

Case 16
    MsgBox "You can drive now!"
Case 18
    MsgBox "You can vote now!"
Case 21
    MsgBox "You can drink wine with
your meals."
Case 65 to 70
    MsgBox "Time to retire and have
fun!"
End Select

```

II. រង្វង់ (Loops)

Loop គឺជា Statement មួយដែលមានសារៈសំខាន់បំផុតនៅក្នុងភាសាសរសេរកម្មវិធីព្រោះវាអនុញ្ញាតអោយយើងដំណើរការ Code មួយបន្តគ្នា ឬក៏ច្រើនបន្តគ្នាដែលខ្លះពិតជាជួបលក្ខខណ្ឌមួយទើបវាឈប់ដំណើរការ ។

នៅក្នុងភាសា Visual Basic Loop Statements មានដូចជា:

1. Do While...Loop
2. Do Until...Loop
3. Do...LoopWhile
4. Do...UntilWhile
5. While...Wend
6. For...Next
7. For Each...Next

1. Do While...Loop:

- វាធ្វើការត្រួតពិនិត្យលក្ខខណ្ឌមុននឹងដំណើរការ Block Loop Statements ។
- ប្រសិនបើលក្ខខណ្ឌនោះពិតនោះរាល់ Block Statement របស់ Loop ត្រូវបានដំណើរការរហូតដល់ជួបលក្ខខណ្ឌមិនពិត ទើបវាចាកចេញ ។

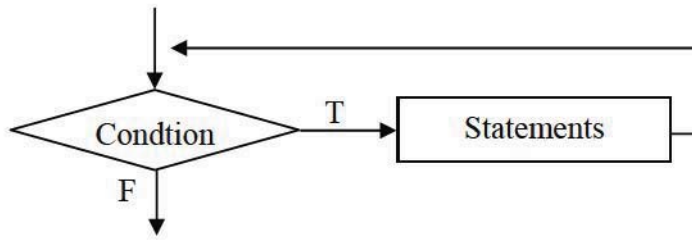
Syntax

```

Do While Condition
    [Statements]
Loop

```


Flow Chart:



Ex:

```

Dim I, Sum as integer
I=1
Do While(I<=100)
    Sum = Sum + 1
    I = I+1
Loop
TxtResult.Text =Sum

```

2. Do Until...Loop:

ចំពោះការប្រើ Do Until...Loop ដំណើរការរបស់វា ដូចគ្នាទៅនឹង Do While...Loop ដែរ តែវាខុសគ្នាត្រង់:

- កាលណាសក្ខីខ័ណ្ឌរបស់ Loop មិនពិតទើប Loop Statements ដំណើរការ
- កាលណាសក្ខីខ័ណ្ឌរបស់ Loop ពិតវិញនោះវានឹងចាកចេញពី Loop

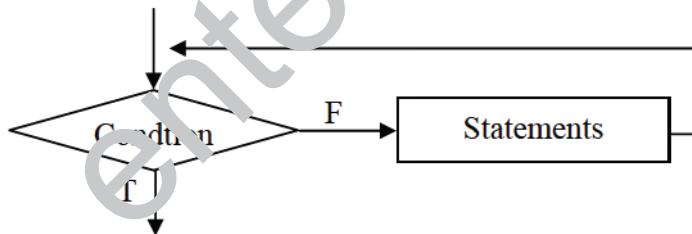
Syntax:

Do Until Condition

[Statements]

Loop

Flow Chart:



Ex:

```

Dim I, Sum as integer
I=1
Do Until(I>100)
    Sum = Sum + 1
    I = I+1
Loop
TxtResult.Text =Sum

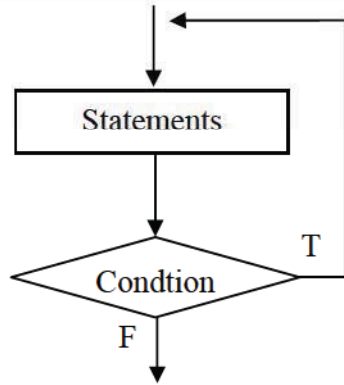
```

3. Do...Loop While:

- វាធ្វើការត្រួតពិនិត្យ Statements មុន ទើបវាត្រួតពិនិត្យសក្ខីខ័ណ្ឌក្រោយ ។
- ប្រសិនបើសក្ខីខ័ណ្ឌនោះពិតនោះរាល់ Block Statement របស់ Loop ត្រូវបានដំណើរការរហូតដល់ដប់សក្ខីខ័ណ្ឌមិនពិតទើបវាចាកចេញ ។

Syntax:
Do
 [Statements]
Loop While Condition

Flow Chart



Ex:

```
Dim I, Sum as integer
I=1
Do
  Sum = Sum + 1
  I = I+1
Loop While(I<=100)
TxtResult.Text =Sum
```

4. Do...Loop Until:

ដំណើរការរបស់គឺ វាធ្វើការត្រួតពិនិត្យ Statements ទុន ទើបវាត្រួតពិនិត្យលក្ខខណ្ឌក្រោយ ។ តែវាខុសគ្នាត្រង់:

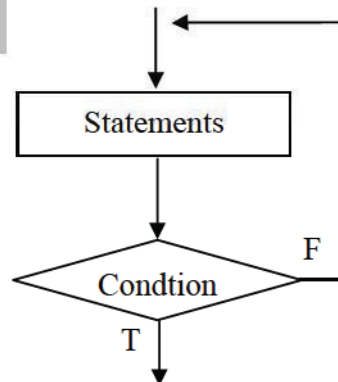
- កាលណាលក្ខខណ្ឌរបស់ Loop មិនពិត ទើប Loop Statements ដំណើរការ
- កាលណាលក្ខខណ្ឌរបស់ Loop ពិត ទើបវាទៅនឹងចាកចេញពី Loop

Syntax:
Do
 [Statements]
Loop Until Condition

Ex:

```
Dim I, Sum as integer
I=1
Do
  Sum = Sum + 1
  I = I+1
Loop Until(I>100)
TxtResult.Text =Sum
```

Flow Chart:



5. While...End While:

ចំពោះការប្រើ While...End While គឺដូចគ្នារបស់ទៅនឹងការប្រើ Do While...Loop ។

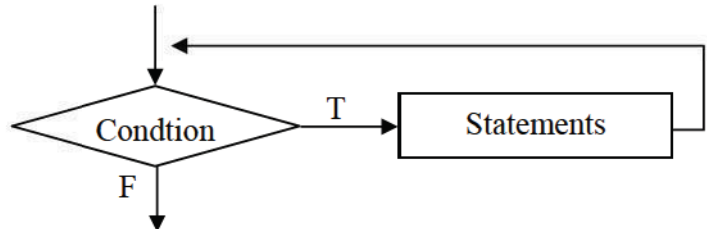
Syntax:

While Condition

[Statements]

End While

Flow Chart:



Ex:

```

Dim I, Sum as integer
I=1
Do While(I<=100)
    Sum = Sum + i
    I = I+1
Loop
TxtResult.Text =Sum

```

6. For...Next

Syntax:

For CounterVariable=Start 1, End [Step increment]

Loop_Statements

Next CounterVariable

-CounterVariable = Start មានន័យថាវាយឹងត្រូវកំណត់តំលៃដំបូងទៅអោយចំនួនលើក ដែល Loop ត្រូវធ្វើ

-End ជាតំលៃបញ្ចប់នៃ Loop

-[Step increment] ជាជំហានដែលត្រូវកើន ឬ ថយចុះជា Default គឺ ១

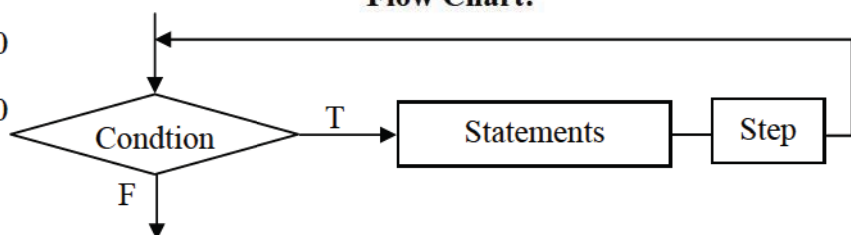
Note:

-បើ Start > End នោះ n < 0

-បើ Start < End នោះ n > 0

Default របស់ Step គឺ 1

Flow Chart:



Ex1:

```

For i = 1 To 100 Step 1
    Sum += i
Next
TxtResult.Text=Sum

```

Ex2:

```

For i = 1 To 100
    Sum=Sum + i
Next
TxtResult.Text=Sum

```

Ex3:

```

For i = 100 to 1 Step -1
    Sum = Sum + i
Next
TxtResult.Text=Sum

```

Using Control Structure in Programming

១- សរសេរ Program ដើម្បីវាយបញ្ចូល Password បានចំនួន ១០ តួអក្សរ ហើយបើវាយខុស ៣ ដង បិទ Form ដោយខ្លួនឯង?



Object	Properties Name	Properties Setting
Label1	Text	&User name
Lable2	Text	&Password
TextBox1	Name	txtUser
TextBox2	Name	txtPassword
Button1	Name	Btnok
	Text	Ok
Button2	Name	Btncancel
	Text	Cancel

>> កំរិតសរសេរ Code:

```
Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btncancel.Click
    Me.Close()
End Sub
```

```
Private Sub txtpassword_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles txtpassword.TextChanged
```

```
    If Len(Me.txtpassword.Text) >= 10 Then
        Me.txtpassword.Text = Mid(Me.txtpassword.Text, 1, 10)
        Me.txtpassword.SelectionStart = 10
    End If
End Sub
```

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

```
    Static i As Byte
    If Trim(UCase(Me.txtpassword.Text)) = "NUM" Then
```



```

MsgBox("Your Password is Correct!")
i = 0
Else
MsgBox("Your Password isn't Correct!")
Me.txtpassword.Focus()
Me.txtpassword.SelectionStart = 0
Me.txtpassword.SelectionLength = Len(Me.txtpassword.Text)
i += 1
End If

If i = 3 Then
Close()
End If
End Sub

```

២-ឧបមាថាវិញមាន Form ដូចខាងក្រោម

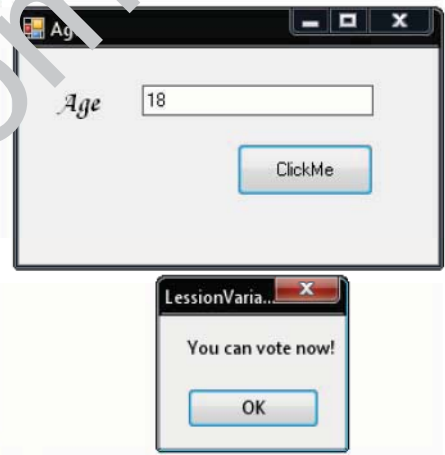
```

Private Sub btnclick_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnclick.Click
If IsNumeric(Me.txtage.Text) = False Then
MsgBox("Invalid Number!")
Me.txtage.Focus()
Me.txtage.SelectionStart = 0
Me.txtage.SelectionLength= Len(Me.txtage.Text)
Exit Sub
Else

If Me.txtage.Text = 16 Then
MsgBox("You can drive now!")
ElseIf Me.txtage.Text = 18 Then
MsgBox("You can vote now!")
ElseIf Me.txtage.Text = 21 Then
MsgBox("You can drink wine with your meals.")
ElseIf Me.txtage.Text = 60 Or Me.txtage.Text = 66 Then
MsgBox("Time to retire and have fun!")

End If
End If
End Sub

```



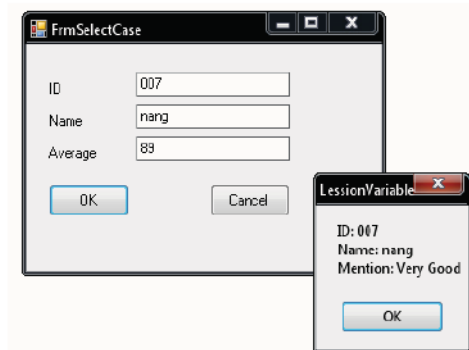
៣-សរសេរ Program ដើម្បីបង្ហាញ Mention របស់សិស្សដូចខាងក្រោម:

```

Private Sub btnok_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnok.Click
Dim Total As Single
Dim str As String
Total = Val(Me.txtavg.Text)

Select Case Total
Case 90 To 100
str = "Excellent"
Case Is >= 80
str = "Very Good"
Case Is >= 70
str = "Good"
Case Is >= 60
str = "Fair"

```



```

Case Is >= 50
str = "Pass"
Case Else
str = "Fail"

End Select
MsgBox("ID: " &
Me.txtid.Text & vbCrLf &
"Name: " & Me.txtname.Text
& vbCrLf & "Mention: " & str)
End Sub

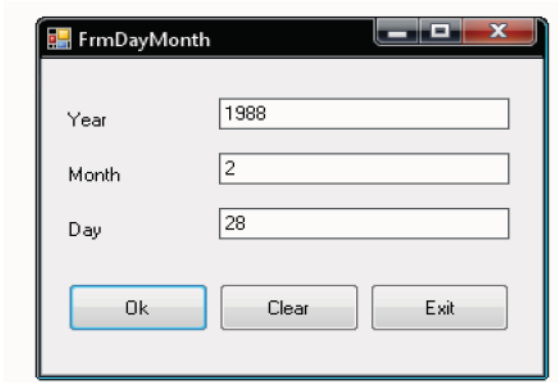
Private Sub
btnCancel_Click(ByVal sender
As System.Object, ByVal e As

```

System.EventArgs) Handles btnCancel.Click
End
End Sub

៤-សរសេរ Program ដើម្បីរកថ្ងៃដាច់នៃខែ:

- បើខែ 1,3,5,7,9,10,12 នោះថ្ងៃដាច់នៃខែគឺថ្ងៃទី 31
- បើខែ 4,6,8,11 នោះថ្ងៃដាច់ នៃខែគឺថ្ងៃទី 30
- បើខែ 2 នោះថ្ងៃដាច់ នៃខែគឺថ្ងៃទី 29 ឬ 28



- ការសរសេរ Code:

Private Sub btnok_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click

If Me.txtmonth.Text = "1" Or Me.txtmonth.Text = "3" Or Me.txtmonth.Text = "5" Or Me.txtmonth.Text = "7"
Or Me.txtmonth.Text = "9" Or Me.txtmonth.Text = "10" Or Me.txtmonth.Text = "12" Then

Me.txtday.Text = "31"

ElseIf Me.txtmonth.Text = "4" Or Me.txtmonth.Text = "6" Or Me.txtmonth.Text = "8" Or Me.txtmonth.Text =
"11" Then

Me.txtday.Text = "30"

ElseIf Me.txtmonth.Text = "2" Then

Dim Y As Integer

Y = Val(Me.txtyear.Text) Mod 4

If Y = 0 Then

Me.txtday.Text = "28"

Else

Me.txtday.Text = "29"

End If

End If

End Sub

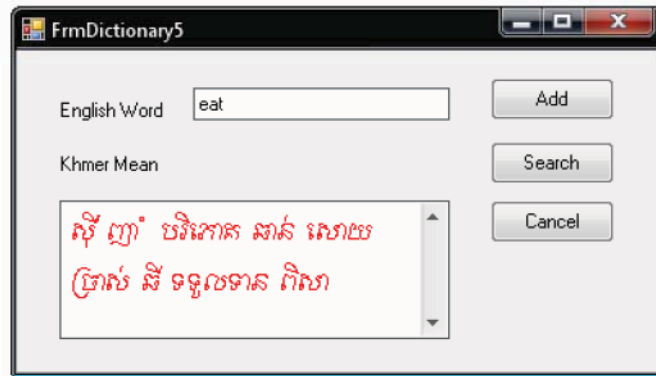
Private Sub btnclear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnclear.Click

Me.txtyear.Clear()

Me.txtmonth.Clear()

```
Me.txtday.Clear()
End Sub
```

៥-សរសេរ Program ដើម្បីបង្កើត Dictionary ផ្ទុកបាន ៥ ពាក្យ:



>>ការសរសេរ Coode:

```
Dim w1, w2, w3, w4, w5 As String
Dim m1, m2, m3, m4, m5 As String
Dim ind As Integer
```

```
Private Sub btnadd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnadd.Click
    If ind = 0 Then
        w1 = Me.txtword.Text
        m1 = Me.txtmean.Text
        ind += 1
    ElseIf ind = 1 Then
        w2 = Me.txtword.Text
        m2 = Me.txtmean.Text
        ind += 1
    ElseIf ind = 2 Then
        w3 = Me.txtword.Text
        m3 = Me.txtmean.Text
        ind += 1
    ElseIf ind = 3 Then
        w4 = Me.txtword.Text
        m4 = Me.txtmean.Text
        ind += 1
    ElseIf ind = 4 Then
        w5 = Me.txtword.Text
        m5 = Me.txtmean.Text
        ind += 1
    Else
        MsgBox("Have no Space!")
    End If
    Me.txtword.Clear()
    Me.txtmean.Clear()
    Me.txtword.Focus()
End Sub
```

Private Sub btnsearch_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsearch.Click

```

    If UCCase(Trim(Me.txtword.Text)) = UCCase(w1) Then
        Me.txtmean.Text = m1
    ElseIf UCCase(Trim(Me.txtword.Text)) = UCCase(w2) Then
        Me.txtmean.Text = m2
    ElseIf UCCase(Trim(Me.txtword.Text)) = UCCase(w3) Then
        Me.txtmean.Text = m3
    ElseIf UCCase(Trim(Me.txtword.Text)) = UCCase(w4) Then
        Me.txtmean.Text = m4
    ElseIf UCCase(Trim(Me.txtword.Text)) = UCCase(w5) Then
        Me.txtmean.Text = m5
    Else
        MsgBox("Have no word!")
    End If
    Me.txtword.Focus()
    Me.txtword.SelectionStart = 0
    Me.txtword.SelectionLength = Len(Me.txtword.Text)

```

End Sub

៦-សរសេរ Program ដើម្បីអាចអោយ User អាចចាក់ឆ្នោតបាន:



>> ការសរសេរ Code:

Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnStart.Click

```

    If Me.btnStart.Text = "Start" Then
        Me.btnStart.Text = "Stop"
        Me.Timer1.Enabled = True
    ElseIf Me.btnStart.Text = "Stop" Then
        Me.btnStart.Text = "Start"
        Me.Timer1.Enabled = False
    If Trim(Me.lblottery.Text) = Trim(Me.txtnum.Text) Then
        Dim Money As Double
        Money = Val(Me.txtmoney.Text) * 10
        Me.lblresult.Text = "You win $" & Money
    Else
        Me.lblresult.Text = "Sorry! you are lose!"
    End If

```



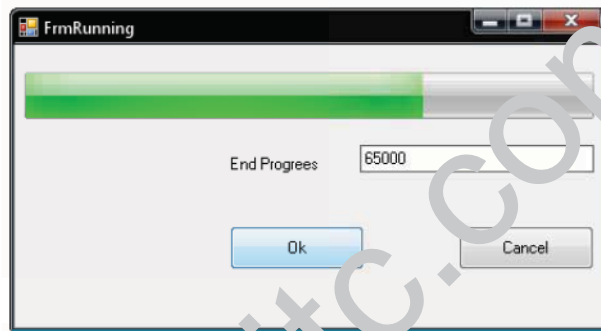
```
End If
End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Me.lblottery.Text = Int(Rnd() * 100)
```

```
End Sub
Private Sub btnclear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnclear.Click
    Me.txtnum.Clear()
    Me.txtmoney.Clear()
End Sub
```

```
Private Sub btnexit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click
    End
End Sub
```

៧-សរសេរ Program ដើម្បីបង្កើត Form Running:



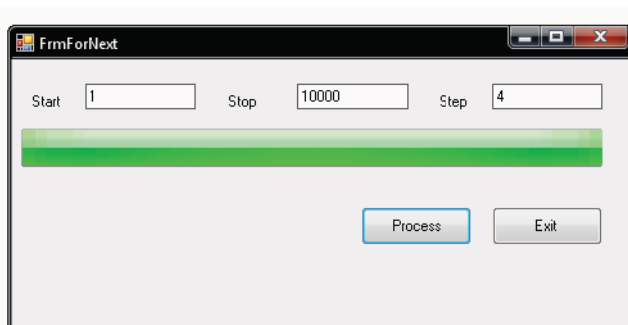
>>កំរសរសេរ Code:

```
Private Sub btnok_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click
    Me.Pro1.Maximum = Val(Me.txtvalue.Text)

    Do Until (Me.Pro1.Value >= Me.txtvalue.Text)
        Me.Pro1.Value += 1
    Loop
End Sub
```

```
Private Sub btncancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btncancel.Click
    End
End Sub
```

៨-សរសេរ Program ដើម្បីបង្កើត Form Running ដោយប្រើ For...Next:





>> ការសរសេរ Code:

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

```
Dim i As Long
```

```
Me.Pro.Maximum = Me.txtstop.Text
```

```
For i = Me.txtstart.Text To Me.txtstop.Text Step Val(Me.txtstep.Text)
```

```
Me.Pro.Value = i
```

```
Next i
```

```
Me.Pro.Minimum = Me.txtstart.Text
```

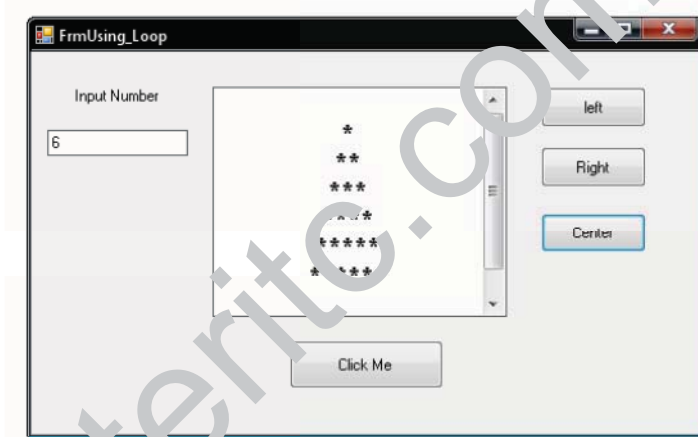
```
End Sub
```

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

```
End
```

```
End Sub
```

៩-សរសេរកម្មវិធីដោយប្រើជា Loop Statement ដើម្បីបោះសញ្ញាផ្តោយអោយមាននៅក្នុង TextBox ដែលមានដូចរូបខាងក្រោម ។



>> ការសរសេរ Code:

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

```
Dim Lp, i As Integer
```

```
Dim str As String = ""
```

```
Lp = Val(Me.txtnumber.Text)
```

```
Do While (i <= Lp)
```

```
For j As Integer = 1 To i
```

```
str &= "*"
```

```
Next
```

```
str &= vbCrLf
```

```
i += 1
```

```
Loop
```

```
Me.txtshow.Text = str
```

```
End Sub
```

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

```
Me.txtshow.TextAlign = HorizontalAlignment.Left
```

```
End Sub
```

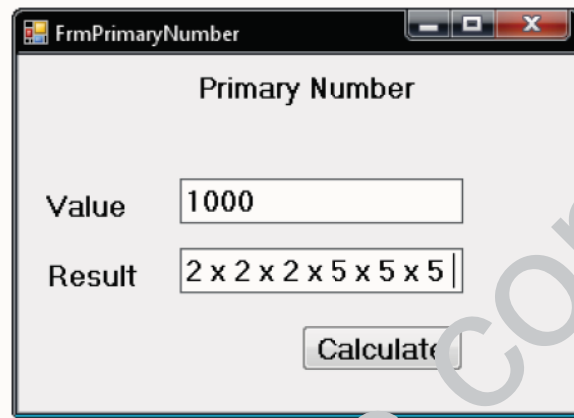
Private Sub Btnright_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btnright.Click

 Me.txtshow.TextAlign = HorizontalAlignment.Right
End Sub

Private Sub btncenter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btncenter.Click

 Me.txtshow.TextAlign = HorizontalAlignment.Center
End Sub

១០-សរសេរកម្មវិធីរកចំនួនបឋម ដូច Form ខាងក្រោម:



>> ការសរសេរ Code:

Private Sub btncalculate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btncalculate.Click

 Dim A, B As Integer
 Dim Str As String

 A = Val(Me.txtnumber.Text)

 Do Until (A < B)

 For B = 2 To A

 If (A Mod B = 0) Then

 A = A / B

 Str &= B & " x "

 Exit For

 End If

 Next

 Loop

 Dim i As Byte

 i = Len(Trim(Str))

 Me.txtprimary.Text = Mid(Str, 1, i - 1)

End Sub

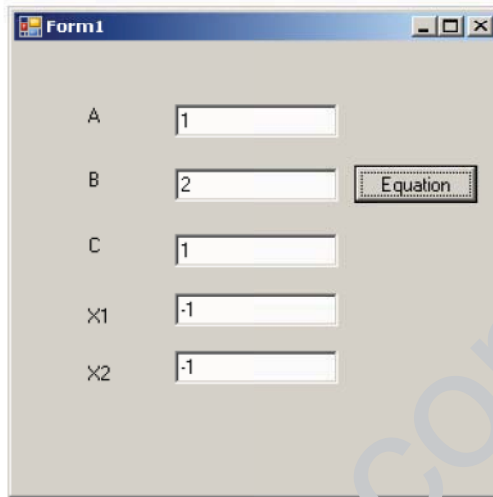
១១-សរសេរកម្មវិធីរករឹសសមីការដឺក្រេទី២ $ax^2 + bx + c = 0$ ($a \neq 0$)

$$\Delta = b^2 - 4ac$$

$\Delta > 0$ មានរឹសពីរ $x_1; x_2 = \frac{-b \pm \sqrt{\Delta}}{2a}$

$\Delta = 0$ មានរឹសមួយ $x_1 = x_2 = -\frac{b}{2a}$

$\Delta < 0$ គ្មានរឹស



Public Class Form1

Private Sub btnequation_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnequation.Click

```

Dim d As Double
Dim x1 As Double
Dim x2 As Double
Dim a As Integer
Dim b As Integer
Dim c As Integer
If txta.Text = "" Then
    MsgBox("Invalid equation degree 2")
    txta.Focus()
Else
    If txta.Text = 0 Then
        MsgBox("Invalid equation degree 2")
        txta.Clear()
        txta.Focus()
        Exit Sub
    End If
    a = Val(txta.Text)
    b = Val(txtb.Text)
    c = Val(txtc.Text)
    d = b ^ 2 - (4 * a * c)
    If d < 0 Then
        MsgBox("Square no root")
        txtx1.Text = "No root"
        txtx2.Text = "No root"
        Exit Sub
    Else
        x1 = (-b + Math.Sqrt(d)) / 2 * a
        x2 = (-b - Math.Sqrt(d)) / 2 * a
    End If
End Sub
    
```

// ខ្លឹមសម្រប

Microsoft
Visual Studio 2008 Administrator Mode

```
txtx1.Text = x1  
txtx2.Text = x2  
End If  
End Sub  
End Class
```

enteritc.com

បេរឿនទី ៦

កម្មវិធីរង (Procedures)

កម្មវិធីរងគឺជាចំណែកតូចនៃកម្មវិធី ឬ ជាកំណត់នៃ Code ដែលធ្វើការផ្តល់បញ្ហាបង្កើតបានជាកម្មវិធីធំ (Application) មួយ ។

កម្មវិធីរង (Procedures) នៅក្នុងភាសា Visual Basic បែងចែកចេញជា ៥ ប្រភេទគឺ:

- Event Procedure
- Function Procedures
- Sub Procedures
- Property Procedure

ម៉ូឌុល

-យើងអាចកំណត់កំរិតដំណើរការរបស់កម្មវិធីរងដោយប្រើ Keyword តែពីរគត់គឺ Public ឬ Private(កំរិតដំណើរការរបស់វាដូចគ្នាទៅនឹងការប្រើប្រាស់ Private ឬ Public នៅក្នុងការប្រកាសអថេរ) ។

-យើងអាចធ្វើការបង្កើតកម្មវិធីរងនៅក្នុង Modules បានទាំងអស់ ។

-ពេលដែលយើងធ្វើការបង្កើតកម្មវិធីរងនៅក្នុង Standard Module ដោយមិនបញ្ជាក់កំរិតដំណើរការជា Public ឬ Private នោះ Compiler នឹងចាត់ទុកកម្មវិធីនោះមានកំរិតដំណើរការជា Public ដោយស្វ័យប្រវត្តិ ។

អត្ថប្រយោជន៍ការបង្កើតកម្មវិធីរង

-ជៀសវាងការសរសេរ Code ដែលមានន័យថា Code ទាំងឡាយណាដែលទទួលបានលទ្ធផលដូចគ្នា មានន័យថា កាត់បន្ថយពេលវេលា ក្នុងការ សរសេរ Code ។

-ងាយស្រួលក្នុងការគ្រប់គ្រង Source Code មានន័យថា វាធ្វើអោយយើងងាយស្រួលក្នុងការត្រួតពិនិត្យមើល ចំនុច Error, ងាយស្រួលក្នុងការកែ Source Code ដើម្បីធ្វើការកែប្រែកម្មវិធី ។

-យើងអាចយកកម្មវិធីរងដែលយើងបានបង្កើត យកទៅប្រើក្នុងកម្មវិធីផ្សេងទៀត ។

I. Sub Procedures

a. និយមន័យ និង Syntax

Sub Procedure ឬហៅថា Subroutine គឺជាកម្មវិធីដែលកើតឡើងពីការផ្តុំគ្នានៃ Statements ជាច្រើនដែល Statements ទាំងនោះបិទនៅក្នុងចន្លោះ Sub និង End Sub Statement ។

Sub Procedure Syntax:

```
[Public/Private] Sub ProcedureName ([argumentList[As Type]])
    Procedure Statement
End Sub
```

សមាសភាពក្នុង Sub Procedure Syntax	អត្ថាធិប្បាយ
Public ឬ Private	ជាកំរិតដំណើរការរបស់ Sub Procedure
ProcedureName	ជាឈ្មោះ Procedure ដែលយើងត្រូវបង្កើត
ArgumentList	ជា Optional arguments ដែលត្រូវបានប្រើនៅក្នុង Sub Procedure ។ ប្រសិនបើ Optional arguments លើសពីមួយយើងត្រូវប្រើសញ្ញា Comma <<,>> ដើម្បីខ័ណ្ឌ រាល់ arguments ទាំងនោះ ។
Procedure Statements	ជាប្រភេទទិន្នន័យដែលយើងត្រូវ ទៅអោយ argumentList
As Type	គឺជាបន្តនៃ Statements ដែលប្រើសំរាប់កំណត់ទៅអោយ Procedure មានតួនាទីជួយ ។
[]	ជា Optional arguments រាល់សមាសភាពដែលនៅក្នុងសញ្ញា [] យើងអាចប្រើកំណត់អត់ក៏បាន ។

ឧទាហរណ៍:

ឧទាហរណ៍យើងបង្កើត Sub Procedure មួយដែលមានឈ្មោះថា Clear ដែលមានតួនាទីសំរាប់លុបតួអក្សរ ដែលមាននៅលើ TextBox ចំនួនបីគឺ (txtValue1, txtValue2, txtResult) ។

— រូបថត ១

ឧបមាថាយើងធ្វើការបង្កើត Procedure ដែលមានឈ្មោះថា Clear ដោយមិនទាមទារអោយមាន Arguments ។

Source Code:

```
Private Sub Clear ()
    txtvalue1.Text= ""
    txtvalue2.Text=""
    txtResult.Text=""
End Sub
```

— រូបថត ២

ឧបមាថាយើងធ្វើការបង្កើត Procedure ដែលមានឈ្មោះថា Clear ដោយទាមទារអោយមាន Arguments វិញ ។

Source Code:

```
Private Sub Clear(txtv1 As TextBox, txtv2 As TextBox, txtR As TextBox)
    txtv1.Text=""
    txtv2.Text=""
    txtR.Text=""
End Sub
```

b. ការហៅ Sub Procedure មកប្រើ

Syntax

```

-ប្រសិនបើមាន Argument
    Call ProcedureName()
រឺ
    ProcedureNameargumentList
-ប្រសិនបើមិនមាន Argument
    Call ProcedureName
រឺ
    ProcedureName

```

🗒 ចំណាំ

-ពេលយើងប្រើនូវ Call KeyWord ដើម្បីហៅ Sub Procedure មកប្រើ នោះរាល់ argument ទាំងអស់របស់ Procedure ត្រូវដាក់ក្នុងសញ្ញាវង់ក្រចក ។

-ប្រសិនបើយើងមិនប្រើនូវ Call KeyWord ក្នុងការហៅ Sub Procedure ទេនោះ រាល់ arguments ទាំងអស់មិនត្រូវដាក់ក្នុងសញ្ញាវង់ក្រចកឡើយ ។

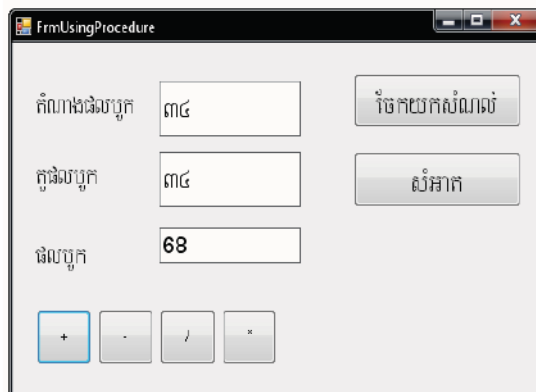
-ប្រសិនបើមាន arguments ហើសតែមួយនោះ ពី argument មួយទៅ argument មួយត្រូវខ្ចីណ្ហដោយសញ្ញា Comma (,) ។

-ប្រសិនបើ Sub Procedure របស់យើងមិនមាន argument ទេ នោះយើងអាចប្រើ Call key Word ក៏បានឬ មិនប្រើក៏បាន ។

-រាល់ argumentList ទាំងអស់ត្រូវមានប្រភេទទិន្នន័យដូចគ្នាទៅនឹង ប្រភេទទិន្នន័យ របស់ ParameterList ។

ឧទាហរណ៍ ១:

ឧបមាថាយើងមានទម្រង់ Form ដូចរូបខាងក្រោម:





Source Code:

```

Public Class FrmUsingProcedure
    Dim V1 As Double
    Dim V2 As Double

.....

Private Sub Clear()
    Me.txtvalue1.Text = ""
    Me.txtvalue2.Clear()
    Me.txtResult.Text = ""
End Sub

.....

Private Sub Sum()
    V1 = Val(Me.txtvalue1.Text)
    V2 = Val(Me.txtvalue2.Text)
    If IsNumeric(Me.txtvalue1.Text) = False Or
IsNumeric(Me.txtvalue2.Text) = False Then

        MsgBox("Invalid Number Please try input again!")
        Exit Sub
    Else
        Me.txtResult.Text = V1 + V2
    End If
End Sub

.....

Sub Subtract()
    V1 = Val(Me.txtvalue1.Text)
    V2 = Val(Me.txtvalue2.Text)
    If IsNumeric(Me.txtvalue1.Text) = False Or
IsNumeric(Me.txtvalue2.Text) = False Then
        MsgBox("Invalid Number Please try input again!")
        Exit Sub
    Else
        Me.txtResult.Text = V1 - V2
    End If

End Sub

.....

Sub Multiply()
    V1 = Val(Me.txtvalue1.Text)
    V2 = Val(Me.txtvalue2.Text)
    If IsNumeric(Me.txtvalue1.Text) = False Or
IsNumeric(Me.txtvalue2.Text) = False Then
        MsgBox("Invalid Number Please try input again!")
        Exit Sub
    Else
        Me.txtResult.Text = V1 * V2
    End If
End Sub

```

Microsoft
Visual Studio 2008 Administrator Mode

```

Sub Modulus ()
    V1 = Val(Me.txtvalue1.Text)
    V2 = Val(Me.txtvalue2.Text)
    If IsNumeric(Me.txtvalue1.Text) = False Or
IsNumeric(Me.txtvalue2.Text) = False Then
        MsgBox("Invalid Number Please try input againt!")
        Exit Sub
    Else
        If Me.txtvalue2.Text = 0 Then
            MsgBox("You can't devide by zero", MsgBoxStyle.Information)
            Exit Sub
        Else
            Me.txtResult.Text = V1 Mod V2
        End If
    End If
End Sub
.....
Sub Divide()
    V1 = Val(Me.txtvalue1.Text)
    V2 = Val(Me.txtvalue2.Text)
    If IsNumeric(Me.txtvalue1.Text) = False Or
IsNumeric(Me.txtvalue2.Text) = False Then
        MsgBox("Invalid Number Please try input againt!")
        Exit Sub
    Else
        If Me.txtvalue2.Text = 0 Then
            MsgBox("You can't devide by zero", MsgBoxStyle.Information)
            Exit Sub
        Else
            Me.txtResult.Text = V1 / V2
        End If
    End If
End Sub
.....
Private Sub FrmUsin_Procedure_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

    Clear()
End Sub
.....
Private Sub btnsum_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsum.Click

    Sum()
End Sub
.....
Private Sub btnSubTraction_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnSubTraction.Click

    Subtract()
End Sub

```

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

```
    Divide()
End Sub
```

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

```
    Multiply()
End Sub
```

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

```
    Modulus()
End Sub
```

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

```
    Clear()
End Sub
End Class
```

ឧទាហរណ៍ ២:

ឧបមាថាយើងធ្វើការកែប្រែរាល់ Sub Procedures

ដែលមាននៅក្នុងឧទាហរណ៍ទី១ ដោយមានទារទារអោយមាន Argument វិញ។

Source Code:

```
Public Class FrmUsingProcedurewithArguments
    Dim V1 As Double, V2 As Double
```

```
Private Sub clear(ByVal txtv1 As TextBox, ByVal txtv2 As TextBox, ByVal txtR As TextBox)
```

```
    txtv1.Text = ""
    txtv2.Text = ""
    txtR.Clear()
End Sub
```



```

Sub Sum(ByVal txtv1 As TextBox, ByVal txtv2 As TextBox, ByVal txtR As
TextBox)
    V1 = Val(txtv1.Text)
    V2 = Val(txtv2.Text)
    If IsNumeric(txtv1.Text) = False Or IsNumeric(txtv2.Text) = False Then
        MsgBox("Invalid Number Please try input again!")
    Else
        txtR.Text = V1 + V2
    End If
End Sub

```

```

Sub Subtract(ByVal txtv1 As TextBox, ByVal txtv2 As TextBox, ByVal
txtR As TextBox)
    V1 = Val(txtv1.Text)
    V2 = Val(txtv2.Text)
    txtR.Text = V1 - V2
End Sub

```

```

Sub Multiply(ByVal txtv1 As TextBox, ByVal txtv2 As TextBox, ByVal
txtR As TextBox)
    V1 = Val(txtv1.Text)
    V2 = Val(txtv2.Text)
    txtR.Text = V1 * V2
End Sub

```

```

Sub Modulus(ByVal txtv1 As TextBox, ByVal txtv2 As TextBox, ByVal txtR
As TextBox)
    V1 = Val(txtv1.Text)
    V2 = Val(txtv2.Text)
    If IsNumeric(txtv1.Text) = False Or IsNumeric(txtv2.Text) = False Then
        MsgBox("Invalid Number Please try input again!")
    Else
        If txtv2.Text = 0 Then
            MsgBox("You can't devide by zero", MsgBoxStyle.Information)
        Else
            txtR.Text = V1 Mod V2
        End If
    End If
End Sub

```

```

Sub Divide(ByVal txtv1 As TextBox, ByVal txtv2 As TextBox, ByVal txtr
As TextBox)
    V1 = Val(txtv1.Text)
    V2 = Val(txtv2.Text)
    If IsNumeric(txtv1.Text) = False Or IsNumeric(txtv2.Text) = False Then
        MsgBox("Invalid Number Please try input again!")
    Else
        If txtv2.Text = 0 Then
            MsgBox("You can't devide by zero", MsgBoxStyle.Information)
        Else

```




```

        txtr.Text = V1 / V2
    End If
End If
End Sub

Private Sub FrmUsingProcedurewithArguments_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Call clear(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub

Private Sub btnsum_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsum.Click
    Call Sum(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub

Private Sub btnSubTraction_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSubTraction.Click
    Subtract(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub

Private Sub btnDivision_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDivision.Click
    Divide(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub

Private Sub btnMultiplication_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMultiplication.Click
    Multiply(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub

Private Sub btnMod_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnMod.Click
    Modulus(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub

Private Sub btnclear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click
    clear(Me.txtvalue1, Me.txtvalue2, Me.txtResult)
End Sub
End Class

```

II. Function Procedures

a. និយមន័យ និង Syntax

Function Procedures គឺជាកម្មវិធីរង (អនុគមន៍) ដែលកើតឡើងការផ្តុំគ្នានៃ Statements ជាច្រើនដែលរាល់



Statements ទាំងនោះប្រើនៅក្នុងចន្លោះ Function និង End Function Statement ហើយកម្មវិធីនេះដើរតួជាអនុគម៌ ដែលរាល់ឈ្មោះរបស់ អនុគម៌ (Function) មានតួនាទីជា អថេរដែលអនុញ្ញាតអោយ យើងបញ្ចូលតំលៃអោយទៅវាបាន។

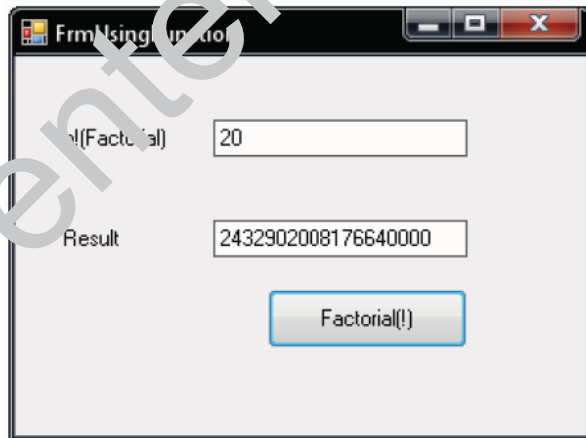
Function Syntax

```
Private/Public Function FunctionName([argumentList[As Type]]) [As Type]
    Function Statements
End Function
```

សមាសភាពដែលមាននៅក្នុង Syntax	អត្ថាធិប្បាយ
FunctionName	ជាឈ្មោះ Function ដែលយើងត្រូវបង្កើត
[As Type]	កំណត់ប្រភេទទិន្នន័យរបស់អោយ argumentsList ឬទៅអោយឈ្មោះ FunctionName មានតួនាទី វាអថេរដែលអាចទទួលយកតំលៃបាន។

ឧទាហរណ៍៖

ចូរបង្កើត Function Procedure មួយសំរាប់គណនាតម្លៃ n factorial(n!) ឧបមាថាយើងមានទម្រង់ Form ដូចខាងក្រោម៖



Source Code:

```
Function Factorial(ByVal n As Long) As Long
    Dim i As Integer
    Dim F As Long
    F = 1
    If n < 0 Then
        MsgBox("n value must be grater than zero", MsgBoxStyle.Information)
    Else
        For i = 1 To n
```



```

F = F * i
Next
End If
Factorial = F " The Same Return F
End Function

```

```

.....
Function Clear() As Integer
    Me.txtnumber.Clear()
    Me.txtResult.Clear()
End Function
.....

```

```

Private Sub btnFact_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnFact.Click
    If Me.txtnumber.Text = "" Then
        MsgBox("Please, enter Integer!!", MsgBoxStyle.Information)
        Exit Sub

    Else
        If IsNumeric(Me.txtnumber.Text) = False Then
            MsgBox("Only Integer,try again")
        Else
            Me.txtResult.Text = Factorial(Me.txtnumber.Text) * CLng(Me.txtnumber.Text)
        End If
    End If
End Sub

```

```

.....
Private Sub btnclear_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnclear.Click
    Clear()
End Sub
.....

```

b. ការហៅ Function Procedure បកប្រែ

Syntax:

- ប្រសិនបើមាន arguments រាល់ arguments ទាំងអស់ត្រូវតែបញ្ចូលនៅក្នុងសញ្ញាវែងក្រចក ។
VariableName=FunctionName(argumentsList)
- ប្រសិនបើគ្មាន Arguments
VariableName=FunctionName រឺ VariableName=FunctionName()

ចំណាំ

-ចំពោះ VariableName ជាចំខាតត្រូវមានប្រភេទទិន្នន័យអោយដូចគ្នាទៅនឹងប្រភេទទិន្នន័យរបស់

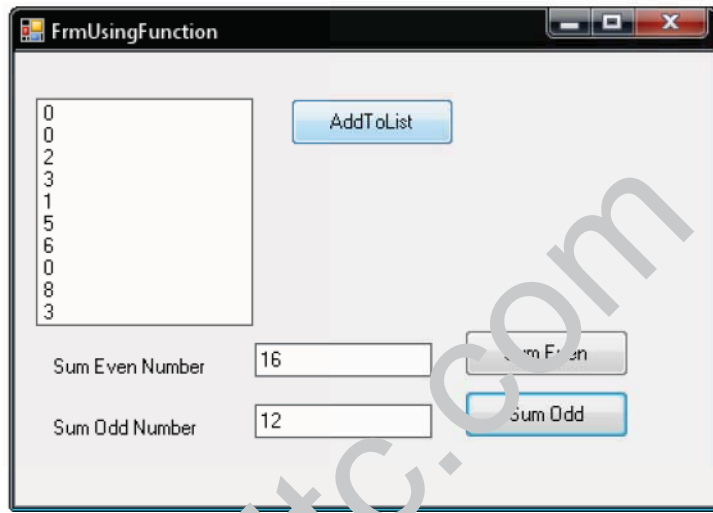
FunctionName ។

-Function Procedure ត្រូវបានប្រើនៅពេលដែលលោកអ្នកធ្វើការគណនានៅចំនួន ណាមួយ ឬក៏ ប្រើដើម្បី ត្រួតពិនិត្យមើលលើលក្ខខណ្ឌអ្វីមួយ ។

-Exit Function វាមានលក្ខណៈស្រដៀងគ្នាទៅនឹង Exit Sub តែ Exit Function ត្រូវបានប្រើជា Function Procedures ។

ឧទាហរណ៍ទី២:

ឧបមាថាយើងមានទម្រង់ Form ដូចរូបខាងក្រោម:



>> ការសរសេរ Code:

```
.....
Sub AddNumberToList(ByVal Lst As ListBox)
    Dim i% " i as integer
    Lst.Items.Clear()
    For i = 1 To 10
        Lst.Items.Add(Int(Rnd() * 10))
    Next
End Sub
```

```
.....
Function SumOddValue(ByVal Lst As ListBox) As Long
    Dim i, SumOdd As Integer
    For i = 0 To Lst.Items.Count - 1
        If Lst.Items.Item(i) Mod 2 <> 0 Then
            SumOdd += CInt(Lst.Items.Item(i))
        End If
    Next
    SumOddValue = SumOdd
End Function
```

```
.....
Function SumEvenValues(ByVal Lst As ListBox) As Long
    Dim i, SumEven As Integer

    For i = 0 To Lst.Items.Count - 1
        If Lst.Items.Item(i) Mod 2 = 0 Then
```




```

SumEven = SumEven + CInt(Lst.Items.Item(i))
End If
Next
Return SumEven
End Function

```

```

Private Sub btnaddlist_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnaddlist.Click
    AddNumberToList(Me.Lst1)
End Sub

```

```

Private Sub btnsumeven_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnsumeven.Click
    Me.txtsumeven.Text = SumEvenValues(Me.Lst1)
End Sub

```

```

Private Sub btnsumodd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnsumodd.Click
    Me.txtOdd.Text = SumOddValue(Me.Lst1)
End Sub

```

III. Using byRef And byVal Keyword

ជាទូទៅនៅពេលដែលយើងធ្វើការបង្កើត Procedure ក្នុងប្រព័ន្ធប្រតិបត្តិការ របស់វាយើងមាន Argument នោះ Compiler ទាមទារអោយលោកអ្នកត្រូវប្រើនូវ byVal ឬ byRef នៅពីមុខរាល់ Arguments ទាំងអស់។ ប្រសិនបើយើងមិនប្រើនូវ Keyword ណាមួយ នោះ Compiler នឹងធ្វើការកំណត់យក byRef Keyword ដោយស្វ័យប្រវត្តិ។

a. Using byRef (by Reference) Keyword

ByRef Keyword គឺមិនរក្សាតំលៃដើមរបស់ arguments ទុកនោះទេព្រោះវាយកតំលៃនោះទៅធ្វើដោយផ្ទាល់

b. Using byVal (by Value) Keyword

ចំពោះ byVal Keyword នៅពេលដែលយើងប្រើវានៅពីមុខរាល់ Arguments នោះតំលៃរបស់ arguments ទាំងអស់ត្រូវបានរក្សាតំលៃដើមទុក ព្រោះពេលដែលដំណើរការ Compiler មិនយក arguments ទៅធ្វើការដោយផ្ទាល់ទេ គឺវាធ្វើការចំលងយកតំលៃទៅធ្វើប៉ុណ្ណោះ។

ឧទាហរណ៍:

```

Private Function Add(ByVal a1 As Integer, ByRef a2 As Integer) As Integer
    Add = a1 + a2
    a1 = 1
    a2 = 1
End Function

```



Private Sub btnadd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnadd.Click

Dim x1, x2, x3 As Integer

x1 = 100

x2 = 200

x3 = Add(x1, x2)

MsgBox("x1=" & x1 & vbCrLf & "x2=" & x2 & vbCrLf & "x3=" & x3)

End Sub

IV. ការប្រើ Arguments ដោយមិនកំណត់ទុកជាមុន

គឺជាការសិក្សាលើការកំណត់តំលៃស្រាប់អោយ Arguments ហើយក្នុងការហៅយកទៅប្រើប្រាស់យើងអាចបោះតំលៃអោយវាក៏បាន អត់ក៏បាន ។ Arguments ដែលយើងមិនបានបោះតំលៃអោយវា គឺវាយកតំលៃដែលមានស្រាប់ ។

ឧទាហរណ៍:

Function rectarea(ByVal b As Double, Optional ByVal a As Double = 1) As Double

rectarea = b * a

End Function

Private Sub btnrectangle_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnrectangle.Click

txtresult.Text = rectarea(txtop1.Text)

End Sub

V. ការប្រើ Arguments ដោយមិនកំណត់ទុកជាមុន

គឺជាការសិក្សាពី របៀបប្រើ Arguments ច្រើននៅក្នុង Procedure មួយដោយមិនកំណត់ទុកជាមុន ។ ដើម្បីធ្វើបែបនេះទៅបាន យើងត្រូវប្រើនូវ Paramarray Keyword ។

Sub Procedure Syntax

```
Private/Public Sub ProcedureName([arguments],[Paramarray ArrayName()])
    Function Statements
End Sub
```

Function Procedure Syntax:

```
Private/Public sub FunctionProcedureName FunctionName ([arguments],[Paramarray ArrayName()])
    Function Statements
End Function
```

ឧទាហរណ៍:

ឧបមាថាយើងធ្វើការបង្កើតនូវ Function Procedure មួយដែលអាចគណនា តំលៃតូចបំផុតនៃពីរចំនួន ឬ បីចំនួន ឬ បួនចំនួន ឬ ដល់ទី n ចំនួនដោយយោងទៅតាមចំនួន Arguments ដែលយើងបោះទៅអោយ Function Procedure ។

Source Code:

```

.....
Function MinValue(ParamArray Value( )) as Double
    Dim i As integer
    Dim Min As Double
    Min=Value(0)
    For i = 1 to UBound(Value)
        If Min > Value(i) Then
            Min=Value(i)
        End If
    Next
    MinValue = Min
End Function

```

សាកល្បងបោះពុម្ពលើកទី១

```

Private Sub Form_Load(...)
    MsgBox MinValue(10,20)
End Sub

```

សាកល្បងបោះពុម្ពលើកទី២

```

Private Sub Form_Load(...)
    MsgBox MinValue(10, 20, ...)
End Sub

```

សាកល្បងបោះពុម្ពលើកទី២

```

Private Sub Form_Load(...)
    MsgBox MinValue(10,20,5,45,60,100,70,-8,-98)
End Sub

```

១-សាកល្បង តំរូវបតំលៃលេខ ពីតូចទៅធំ រឺពីធំមកតូច ២- សាកល្បង បង្កើតអនុគមន៍ផលបូក

បេឡេនី ៧

Array

I. និយមន័យ:

- Array ជា Series នៃ Variables ដែលត្រូវបានប្រើឈ្មោះតែមួយ និង មាន Index(number) សំរាប់កំណត់ធាតុ (Element)
- ធាតុនីមួយៗនៃ Array មានប្រភេទ DataType តែមួយ
- រាល់ធាតុរបស់ Array ទាំងអស់ត្រូវមាន Lower និង Upper Bound ។

ឧ:

	A
0	
1	
2	
3	
4	
5	

សំគាល់:

- Lower Bound : ជា Index ដំបូងគេ របស់ Array
- Upper Bound : ជា Index ចុងក្រោយបំផុតនៃ Array

នៅក្នុង Visual Basic មាន Array ប្រភេទខាងក្រោម:

- Fixed-Side Array
- Dynamic Array
- Jagged Array

II. Fixed-Side Array:

គឺជាការកំណត់ចំនួនធាតុរបស់វាមិនអាចប្រែប្រួល at run-time បានឡើយមុននឹងប្រើវាគេត្រូវការកំណត់ចំនួនធាតុរបស់វាជាមុនសិន ។

II. -1 ការកំណត់ Upper and Lower Bound:

Syntax:

Keyword ArrayName (UpperBound) As Type

+ Keyword មាន Dim,Private,Static,Public

ឧ: Dim MyArray(3) As String



```
MyArray(0)="Sim Selaphean"
MyArray(1)="Kim Saorin"
MyArray(3)="Teary"
```

ទំរង់តារាងអនុក្រុម Memory

Index	MyArray
0	Sim Selaphean
1	Kim Saorin
2	
3	Teary

-MyArray.GetLowerBound(0) ទទួលយកតំលៃ Index តូចបំផុតរបស់ Array មួយវិមាត្រ ។

-MyArray.GetUpperBound(0) ទទួលយកតំលៃ Index ធំបំផុតរបស់ Array មួយវិមាត្រ ។

ឧទាហរណ៍:

```
Dim MyArray( ) As String={"Phean","Teary","Chea Heng"}
```

ទំរង់តារាងអនុក្រុម Memory

Index	MyArray
0	Phean
1	Teary
2	Chea Heng

II. 2- **ទំរង់ប្រកាស** Array:

គេអាចប្រកាស Array ដូចខាងក្រោម

ក-Create Public Array:

ប្រើ Keyword Public ហើយ declare វានៅក្នុង General Declaration នៃ Standard Model ។

```
Public A(10) As String
```

ខ-Create Module-Level Array:

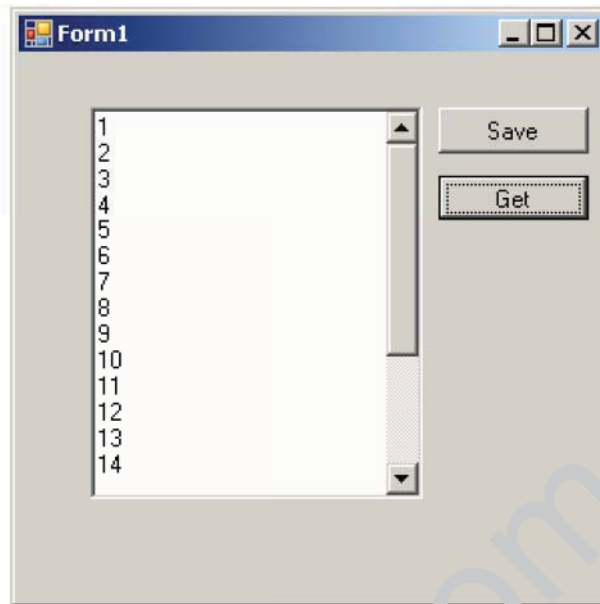
ប្រើ Keyword Private រឺ Dim ហើយ Declare វានៅក្នុង General Declaration នៃ Standard Module ។

```
Private B(4) As String
Private C(6) As Integer
```

គ-Create Local Array:

ប្រើ Keyword Dim រឺ Static ក្នុង Procedure ។

```
Static D(5) As String
```



```
Public Class Form1
    Dim a(19) As Integer
    Dim s As String = ""
    Private Sub btnbinary_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click
        Dim i As Integer
        For i = 0 To 19
            a(i) = i + 1
        Next
    End Sub

    Private Sub btnoctal_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btngce.Click
        Dim i As Integer
        For i = 0 To 19
            s = s & a(i) & ", " & vbCrLf
            txtdisplay.Text = s
        Next
        //txtdisplay.Text = s
    End Sub
End Class
```

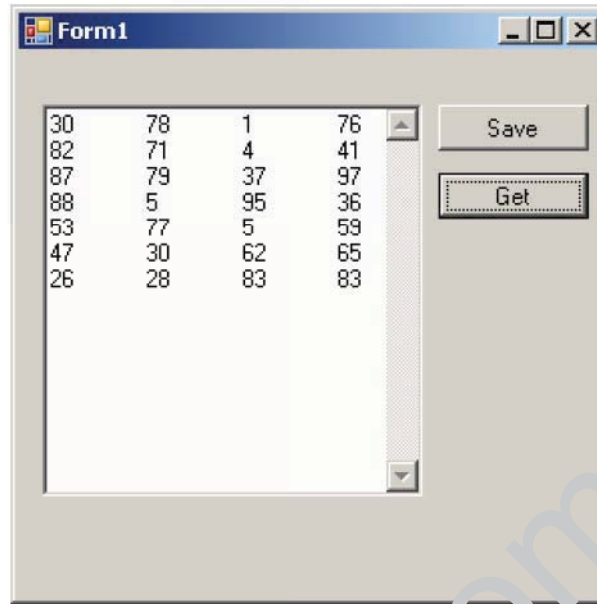
II. 3- Array ប្រើនិមាត្រ

ចំពោះ Array ប្រើនិមាត្រទំរង់នៃការរក្សាទុកទិន្នន័យរបស់វាគឺមានលក្ខណៈដូចគ្នាទៅនឹង Matrix ។

ឧ៖ Dim twodiarray(,) As String={{“Lim Kevtheara”, “1980”, “M”},{“Chea Phalla”, “1982”, “F”}}

ទំរង់របស់វារក្សាទុកនៅក្នុង Memory គឺ៖

RowIndex	ColumnIndex		
	0	1	2
0	Lim Kevtheara	1980	M
1	Chea Phalla	1982	F



```
Public Class Form1
    Dim a(7, 4) As Short
    Dim s As String = ""
    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click
        Dim i, j, k As Short
        For i = 0 To 7
            For j = 1 To 4
                k = k + 1
                a(i, j) = Fix(Rnd() * 101)
            Next
        Next
    End Sub

    Private Sub btnget_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnget.Click
        Dim i, j As Short
        For i = 1 To 7
            For j = 1 To 4
                s = s & a(i, j) & vbTab
            Next
            s = s & vbCrLf
        Next
        txtdisplay.Text = s
    End Sub
End Class
```

III. Dynamic Array:

ជូនពេលខ្លះការប្រកាសអថេរជាប្រភេទ Array ជាប្រភេទ Dynamic Array ព្រោះថាសេរីអ្នកមិនអាចស្មានទុកជាមុនបាននូវទំហំធំបំផុតរបស់ index បាននោះទេ ។

Syntax:

```
Dim ArrayName() As DataType
```

ឧ៖



Dim Myarray(.) As String

ចំពោះការប្រើ Dynamic Array ពេលប្រកាសគេមិនប្រើ uboundindex នោះទេពីព្រោះ uboundindex ត្រូវបានប្រើនៅពេលប្រកាស array ឡើងវិញ។ ដើម្បីប្រកាស Array ឡើងវិញលោកអ្នកអាចប្រើ keyword ពីរគឺ Redim ឬ Redim Preserve ។

III. 1. ការប្រើ Redim Keyword:

Syntax:

Redim ArrayName(uboundindex)

ចំណាំ: ចំពោះការប្រើ Redim Keyword គឺវាមិនអាចរក្សាតំលៃនៃធាតុរបស់ Array បានឡើយព្រោះនៅពេលដំណើរការ ជួប Redim Keyword ពេលនោះបណ្តាលអោយ index របស់ array ត្រលប់មកកាន់ទីតាំងដើម (LBound) វិញដែលជាមូលហេតុធ្វើអោយតំលៃនៃធាតុរបស់វាបាត់បង់។

ឧ: Source Code:

```

Public Class Form1
    Dim a() As String
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ReDim a(2)
        a(0) = "ABC"
        a(1) = "Angkor"
        a(2) = "7 UP"
        MsgBox("a(0)=" & a(0) & vbCrLf & "a(1)=" & a(1) & vbCrLf & "a(2)=" & a(2))
    End Sub
End Class

```

III. 2. ការប្រើ Redim Preserve Keyword:

ចំពោះការប្រើ Redim Preserve Keyword គឺវាអាចរក្សាតំលៃនៃធាតុរបស់ array បានព្រោះនៅពេលដែលជួប keyword redim preserve ពេលនោះ index របស់វារត់មកកាន់ទីតាំងចុងក្រោយហើយទើបបន្ថែម location បន្តទៀត។

ឧ: Source Code:

```

Public Class Redimpreserve
    Dim a() As String
    Private Sub Redimpreserve_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ReDim a(2)
        a(0) = "ABC"
        a(1) = "Angkor"
        a(2) = "Anchor"
        MsgBox("A(0)=" & a(0) & vbCrLf & "A(1)=" & a(1) & vbCrLf & "A(2)=" & a(2))
        ReDim Preserve a(4)
        a(3) = "Tiger"
        a(4) = "Wine"
    End Sub
End Class

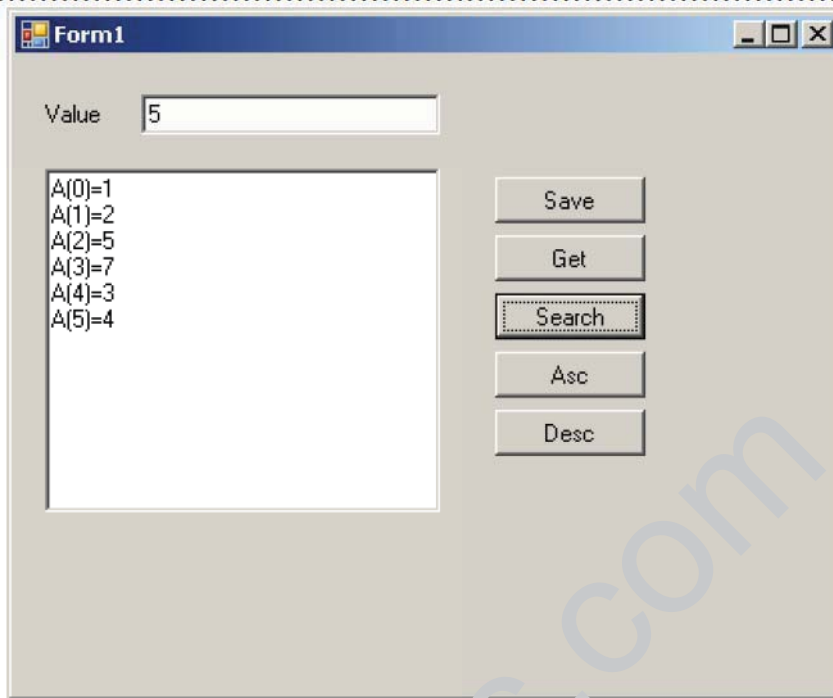
```



```

MsgBox("A(0)=" & a(0) & vbCrLf & "A(1)=" & a(1) & vbCrLf & "A(2)=" & a(2) & vbCrLf
& "A(3)=" & a(3) & vbCrLf & "A(4)=" & a(4))
End Sub
End Class

```



```

Public Class Form1
    Dim a() As Integer
    Dim i As Integer
    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsave.Click
        ReDim Preserve a(i)
        a(i) = Val(txtvalue.Text)
        i = i + 1
        txtvalue.Clear()
        txtvalue.Focus()
    End Sub

    Private Sub btnget_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnget.Click
        Dim j As Integer
        txtdis.Text = ""
        For j = 0 To i - 1
            txtdis.Text = txtdis.Text & "A(" & j & ")=" & a(j) & vbCrLf
        Next
    End Sub

    Private Sub txtvalue_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtvalue.KeyPress
        If Asc(e.KeyChar) <> 8 Then
            If (Asc(e.KeyChar) >= 48 And Asc(e.KeyChar) <= 57) Or Asc(e.KeyChar) = 46
And InStr(txtvalue.Text, ".") <= 0 Then
                Else
                    e.KeyChar = ""
                End If
            End If
        End Sub

```



```

Private Sub btnsearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsearch.Click
    Dim j As Integer
    If txtvalue.Text = "" Then
        MsgBox("Must input value")
        txtvalue.Focus()
        Exit Sub
    End If
    For j = 0 To i - 1
        If a(j) = Val(txtvalue.Text) Then
            MsgBox("A(" & j & ")=" & a(j))
            Exit Sub
        End If
    Next
    MsgBox("Was not found")
    txtvalue.Focus()
End Sub

Private Sub btnasc_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnasc.Click
    Dim j As Integer
    Dim t As Integer
    Dim h As Integer
    For j = 0 To i - 1
        For h = j + 1 To i - 1
            If Val(a(j)) > Val(a(h)) Then
                t = a(j)
                a(j) = a(h)
                a(h) = t
            End If
        Next
    Next
End Sub

Private Sub btndesc_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndesc.Click
    Dim j As Integer
    Dim t As Integer
    Dim h As Integer
    For j = 0 To i - 1
        For h = j + 1 To i - 1
            If Val(a(j)) < Val(a(h)) Then
                t = a(j)
                a(j) = a(h)
                a(h) = t
            End If
        Next
    Next
End Sub
End Class
    
```

Jaged Array:

គឺជា array ដែលនៅក្នុង array ។

Syntax:

Keyword A() As Datatype

Ex:

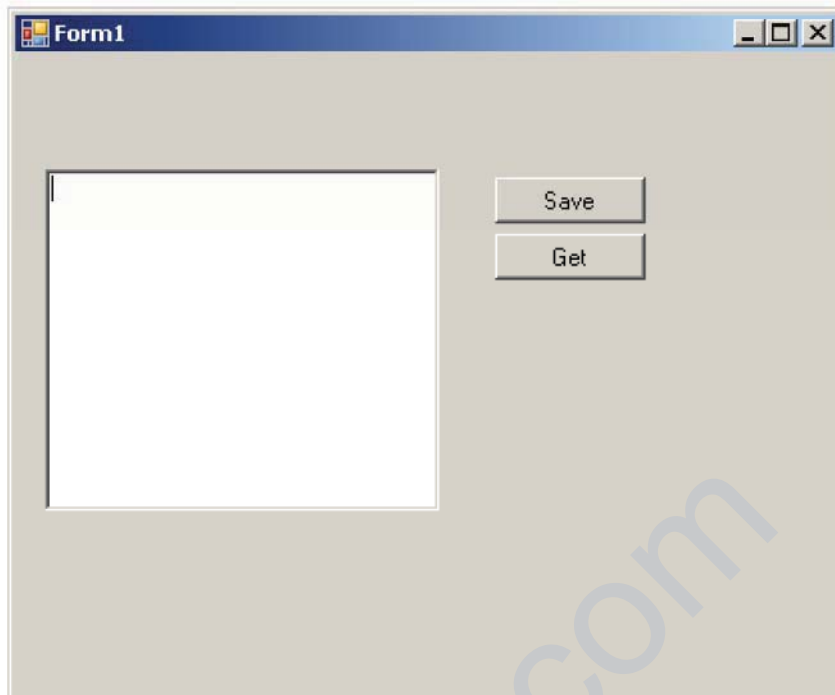
Dim A() () As String

Redim A(3)

A(0) =New String(1){}

A(1) =New String(2){}

☞ Source Code:



```

Public Class jaged
Dim A() As Short
Private Sub btnsave_click(Byval sender as system.object,Byval e as system.EventArgs) Handles
btnsave.click
    Redim A(2)
    A(0)=new short(1){}
    A(1)=new short(2){}
    A(2)=new short(4){}
    Dim i,s,k as short
    for i=0 to 2
        for j= 0 to Ubound(A(i))
            k=k+1
            A(i)(j)=k
        Next
    Next
End Sub
Private Sub btnget_click(Byval sender as system.object,Byval e as system.EventArgs) Handles
btnget.click
    Dim i, j As Short
    Dim s As String
    For i = 0 To 2
        For j = 0 To UBound(A(i))
            s = s & A(i)(j) & vbCrLf
        Next
    Next
    TextBox2.Text = s
End Sub
End Class
    
```

បេឡេនី ៨

FILE SYSTEM OBJECT

I. File System Object:

គឺជា object មួយដែលប្រើសំរាប់គ្រប់គ្រងទៅលើចំនួន Drive, Folder និង file ដែលមាននៅលើ window ។ ដើម្បីប្រើ file system object ត្រូវប្រើកាស Name Space :

Imports System.IO

Method:

1-វេឡេនី Copy File:

Syntax:

File.Copy(Source, Destination)

Ex:

File.Copy("c:\ubb.txt", "d:\ubb.txt")

2-វេឡេនី Move File:

Syntax:

File.Move(Source, Destination)

Ex :

File.Move("c:\ubb.txt", "d:\ubb.txt")

3-វេឡេនី បំបាត់ File:

Syntax:

File.Delete(Path)

Ex:

File.Delete("d:\ubb.txt")

4-វេឡេនី ប្តូរឈ្មោះ File:

Syntax:

Rename(Oldname, Newname)

Ex:

Rename("c:\ubb.txt", "c:\cnit.txt")

5-វេឡេនី ត្រួតពិនិត្យទៅករ File:

Syntax:

File.Exists(Path)

Ex:

File.Exists("c:\ubb.txt")

6-វេឡេនី បង្កើត Folder:

Syntax:

Directory.CreateDirectory(Path)

Ex:



```
Directory.CreateDirectory("c:\ubb")
```

7-វិធីសាស្ត្រ Move Folder:

Syntax:

```
Directory.Move(oldpath,Newpath)
```

Ex:

```
Directory.Move("c:\ubb", "c:\hsx\ubb")
```

8-វិធីសាស្ត្រ Delete Folder:

Syntax:

```
Directory.Delete(Path)
```

Ex:

```
Directory.Delete("c:\hsx\ubb")
```

9-វិធីសាស្ត្រ Exist Folder:

Syntax:

```
Directory.Exist(Path)
```

Ex:

```
Directory.Exist("c:\hsx\ubb")
```

10-វិធីសាស្ត្រ Rename Folder:

Syntax:

```
Rename(Oldname,Newname)
```

Ex:

```
Rename("c:\ubb", "c:\cnit")
```

II. Folder Object:

1-Get Directory:

Syntax:

```
Keyword Variable Array As Data_Type  
Variable = Directory.GetDirectories(Path)
```

Ex:

```
Dim x() As String  
Dim str As String = ""  
x = Directory.GetDirectories("c:\hsx")  
For i = 0 To UBound(x)  
    str = str & x(i) & vbCrLf  
Next
```

2-Get File:

Syntax:

```
Keyword Variable Array As Data_Type  
Variable = Directory.GetFiles(Path,Search Parent)
```

Ex:

```
Dim x() As String  
Dim str As String = ""  
x = Directory.GetFiles("c:\hsx", "*.txt")  
For i = 0 To UBound(x)  
    str = str & x(i) & vbCrLf  
Next
```



2-Get File System Entries:

Syntax:

Keyword Variable_Array As Data_Type
Variable = Directory.GetFilesEntries (Path,Search Parent)

Ex:

```
Dim x() As String
Dim str As String = ""
x = Directory.GetFilesEntries("c:\hsx")
For i = 0 To UBound(x)
    str = str & x(i) & vbCrLf
Next
```

2-Get Logical Drive:

Syntax:

Directory.GetLogicalDrives()

Ex:

```
Directory.GetDirectoryRoot("c:\hsx\avi")
```

III. Collection:

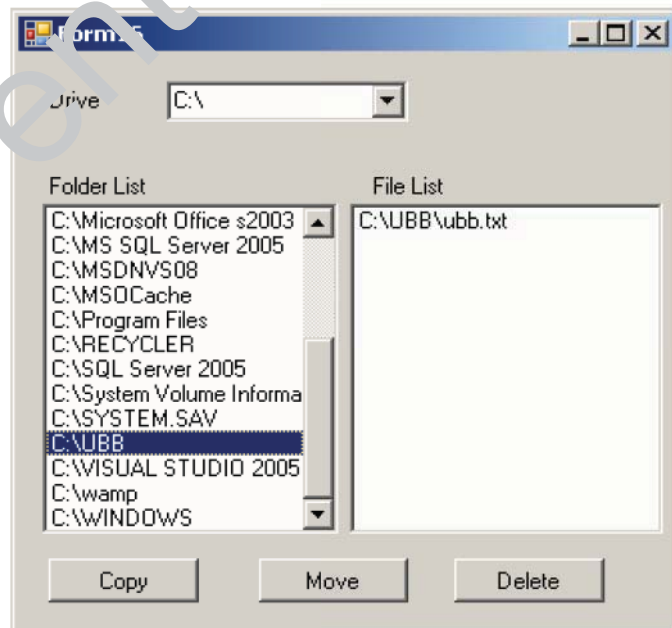
Syntax:

Keyword Variable As Datatype
For Each variable in collection
Statement
Next

Ex:

```
Dim x() As String
Dim s As String
Dim st As String = ""
x = Directory.GetDirectories("c:\hsx")
For Each s In x
    st = st & s & vbCrLf
Next
MsgBox(st)
```

Ex1:





```
Imports System.IO
Public Class Form16
    Dim x() As String
    Dim i As Short
    Private Sub Form16_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        x = Directory.GetLogicalDrives()
        For i = 0 To UBound(x)
            cbodrive.Items.Add(x(i))
        Next
    End Sub

    Private Sub cbodrive_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cbodrive.SelectedIndexChanged
        On Error GoTo a
        lstfolder.Items.Clear()
        lstfile.Items.Clear()
        x = Directory.GetDirectories(cbodrive.Text)
        For i = 0 To UBound(x)
            lstfolder.Items.Add(x(i))
        Next
        x = Directory.GetFiles(cbodrive.Text)
        For i = 0 To UBound(x)
            lstfile.Items.Add(x(i))
        Next
        txtdestination.Text = cbodrive.Text
        btncopy.Enabled = True
        btnmove.Enabled = True
        Exit Sub
a:
        MsgBox("No disk")
        Exit Sub
    End Sub

    Private Sub lstfolder_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lstfolder.SelectedIndexChanged
        lstfile.Items.Clear()
        x = Directory.GetFiles(lstfolder.Text)
        For i = 0 To UBound(x)
            lstfile.Items.Add(x(i))
        Next
        txtdestination.Text = lstfolder.Text
        btnmove.Enabled = True
    End Sub

    Private Sub lstfile_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lstfile.SelectedIndexChanged
        btncopy.Enabled = True
    End Sub

    Private Sub btncopy_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btncopy.Click
        Static b As Boolean
        If b = True Then
            Dim f() As String
            f = Split(txtsource.Text, "\")
            File.Copy(txtsource.Text, txtdestination.Text & "\" & f(UBound(f)))
            btncopy.Text = "Copy"
            btncopy.Enabled = False
            lstfile.Enabled = True
        Else
            txtsource.Text = lstfile.Text
        End If
    End Sub
End Class
```

Microsoft
Visual Studio 2008 Administrator Mode

```

        lstfile.Enabled = False
        btncopy.Text = "Paste"
        btncopy.Enabled = False
    End If
    b = Not b
End Sub

Private Sub btnmove_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnmove.Click
    Static b As Boolean
    Dim f() As String
    f = Split(txtsource.Text, "\")
    If b = True Then
        If InStr(txtsource.Text, ".") > 0 Then
            File.Move(txtsource.Text, txtdestination.Text & "\" & f(UBound(f)))
        Else
            Directory.Move(txtsource.Text, txtdestination.Text & "\" &
f(UBound(f)))
        End If
        lstfile.Enabled = True
        btnmove.Text = "Move"
        btnmove.Enabled = False
    Else
        If lstfile.Text <> "" Then
            txtsource.Text = lstfile.Text
        Else
            txtsource.Text = lstfolder.Text
        End If
        lstfile.Enabled = False
        btnmove.Text = "Paste"
        btnmove.Enabled = False
    End If
    b = Not b
End Sub

Private Sub btndelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndelete.Click
    If lstfile.Text <> "" Then
        File.Delete(lstfile.Text)
        If lstfolder.Text <> "" Then
            lstfolder_SelectedIndexChanged(sender, e)
        Else
            cbodrive_SelectedIndexChanged(sender, e)
        End If
    Else
        Directory.Delete(lstfolder.Text)
        cbodrive_SelectedIndexChanged(sender, e)
    End If

End Sub
End Class

```


បេរៀនទី ៩

ADO.Net

ADO.Net គឺជា object ប្រើសំរាប់ធ្វើការជាមួយ database ។

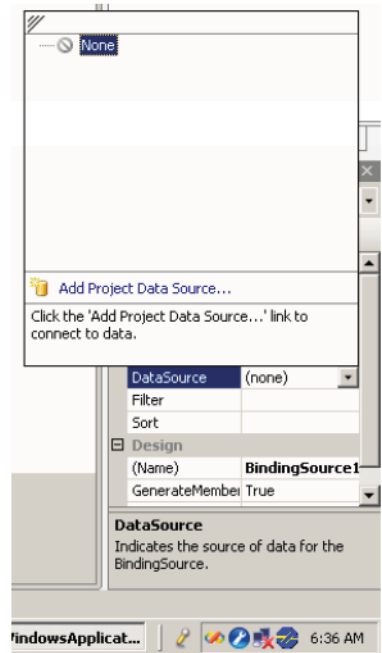
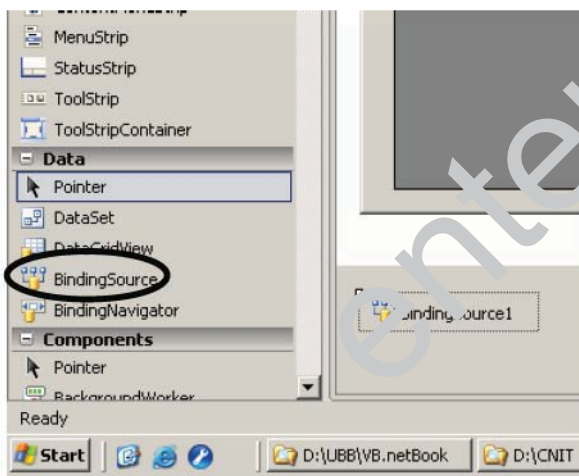
I- ភ្ជាប់ Connection ពី Form ទៅកាន់ DataSource ប្រើ BindingSource

មុននឹងលោកអ្នកចង់ធ្វើការជាមួយនិងទិន្នន័យដែលមាននៅក្នុង DataSource ដូចជាទាញយកទិន្នន័យមកបង្ហាញលើ Controls ផ្សេងៗ,បញ្ចូលទិន្នន័យទៅកាន់ DataSource កែទិន្នន័យដែលមានក្នុង DataSource ជាជំហានដំបូងត្រូវធ្វើការបង្កើត connection ទៅកាន់ពួកវាជាមុនសិន ។

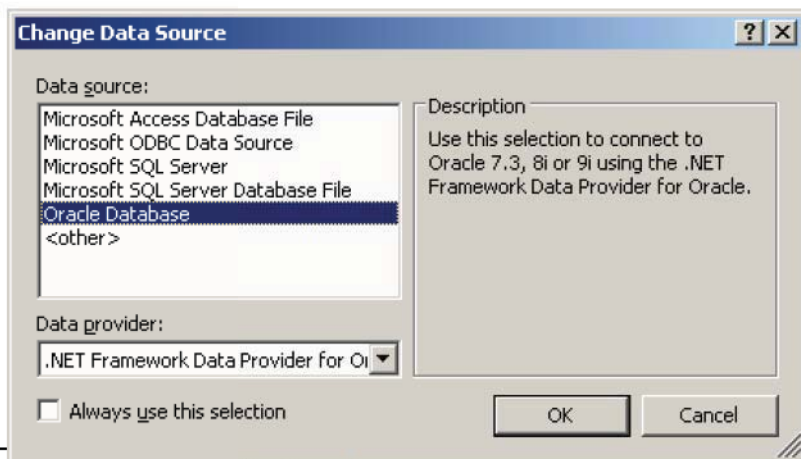
Ex:

1-Binding Source:

- គូស Binding Source Control នៅលើ Form
- ម៉ូស៊ី Mouse លើវា
- កំណត់ DataSource នៅក្នុង Properties



- Add Project Data Source
- Database→Next
- New Connection
- Change



-ជ្រើសរើស Connection អ្វីមួយ

ឱ-Microsoft Access Database File

-Ok→Browse→ជ្រើសរើស database file→Open→Ok

-Next→No→Next

-ជ្រើសរើស Table →Finish

2- ភ្ជាប់ Textbox ជាមួយ DataSource:

-ប្រើ mouse លើ textbox

-ត្រង់ Text របស់ DataBinding

-រើស field ណាមួយ

3- ភ្ជាប់ Binding Navigator Control ទៅកាន់ DataTable:

-ដូស Binding Navigator Control នៅលើ form

-ប្រើ mouse លើ Binding Navigator Control

- ត្រង់ DataSourceនៅក្នុង properties

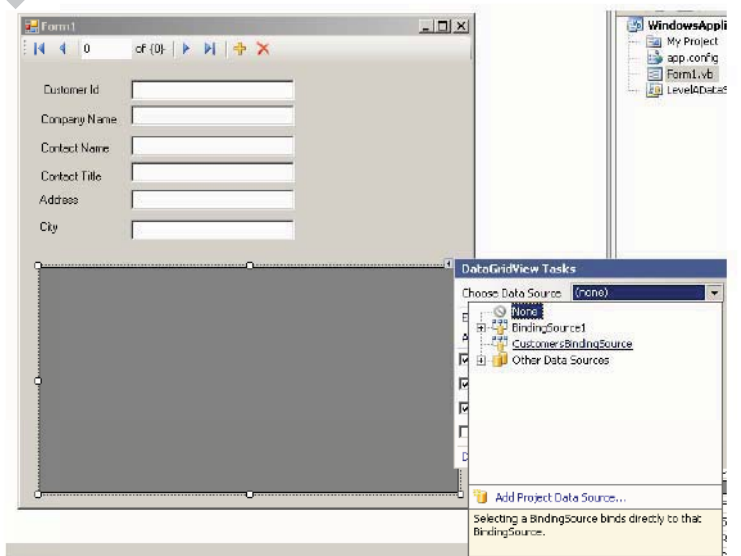
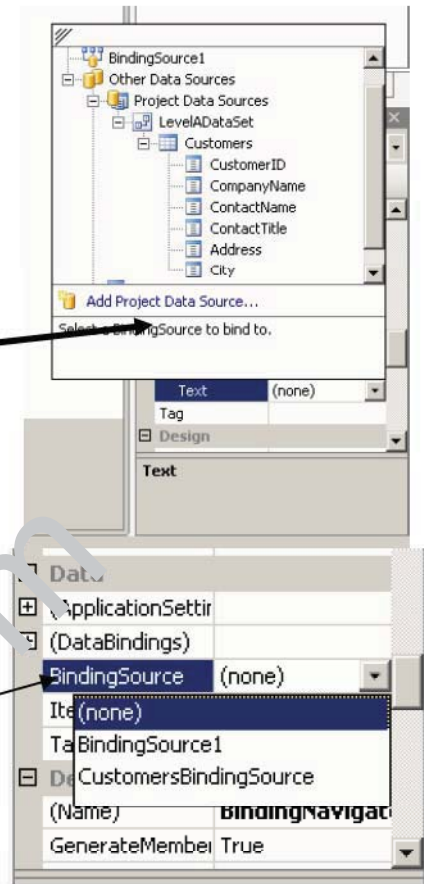
- ជ្រើសរើស table

4- ភ្ជាប់ Datagrid View ជាមួយ Table:

-ដូស Datagrid View Control នៅលើ form

-ប្រើ mouse លើ Data grid View Control

-ជ្រើសរើស table ត្រង់ choose datasource



Ex:

CustomerID	CompanyName	ContactName	ContactTitle
ANTON	Antonio Moreno ...	Antonio Moreno	Owner
AROUT	Around the Horn	Thomas Hardy	Sales Represent
BERGS	Berglunds snabb...	Christina Berglund	Order Administrat
BLAUS	Blauer See Delik...	Hanna Moos	Sales Represent
BLONP	Blondel père et fils	Frédérique Citeux	Marketing Mana
BOLID	Bólido Comidas p...	Martín Sommer	Owner

```
Imports System.Data.OleDb
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'LevelADataset.Customers' table.
        You can move, or remove it, as needed.
        Me.CustomersTableAdapter.Fill(Me.LevelADataset.Customers)
    End Sub

    Private Sub BindingNavigatorDeleteItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BindingNavigatorDeleteItem.Click
        CustomersTableAdapter.Delete(txtid.Text, txtcompany.Text, txtcontact.Text, txttitle.Text, txtaddress.Text, txtcity.Text)
    End Sub

    Private Sub BindingNavigatorAddNewItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BindingNavigatorAddNewItem.Click
        On Error Resume Next
        CustomersTableAdapter.Insert(txtid.Text, txtcompany.Text, txtcontact.Text, txttitle.Text, txtaddress.Text, txtcity.Text)
    End Sub

    Private Sub DataGridView1_CellValueChanged(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellValueChanged
        Static i As Short
        i = i + 1
        If i > 6 Then
            Dim cn As New OleDbConnection
            Dim cm As New OleDbCommand
```




```

cn.ConnectionString = "Provider=Microsoft.Jet.OleDb.4.0;Data
Source=D:\UBB\VB.netBook\test\levela.mdb"
cn.Open()
cm.Connection = cn
cm.CommandType = CommandType.Text
cm.CommandText = "Update customers set CompanyName='" &
DataGridView1.Item(1, e.RowIndex).Value & "',ContactName='" & DataGridView1.Item(2,
e.RowIndex).Value & "',ContactTitle='" & DataGridView1.Item(3, e.RowIndex).Value &
"',Address='" & DataGridView1.Item(4, e.RowIndex).Value & "',City='" &
DataGridView1.Item(5, e.RowIndex).Value & "' where CustomerID='" &
DataGridView1.Item(0, e.RowIndex).Value & "'"
cm.ExecuteNonQuery()
cn.Close()
txtcompany.Text = DataGridView1.Item(1, e.RowIndex).Value
txtcontact.Text = DataGridView1.Item(2, e.RowIndex).Value
txttitle.Text = DataGridView1.Item(3, e.RowIndex).Value
txtaddress.Text = DataGridView1.Item(4, e.RowIndex).Value
txtcity.Text = DataGridView1.Item(5, e.RowIndex).Value
End If
End Sub
End Class

```

II- របៀបភ្ជាប់ Connection ដោយប្រើទូរ OleDbConnection:

OleDbConnection គឺជា Class Object មួយដែលត្រូវបានប្រើសម្រាប់ រក connection ទៅកាន់ datasources ផ្សេងៗដូចជា Text File, Spreadsheet File និង Database File ។ OleDbConnection ជា class ដែលស្ថិតនៅក្នុង System.data.OleDb namespace ។

1-របៀបប្រើ Connection ទៅកាន់ Ms Access:

Syntax:

```

Keyword Variable_object As New OleDb.OleDbConnection
Variable_object.ConnectionString="Provider=Microsoft.Jet.OleDb.4.0;Data
Source=File&Path"
Variable_object.Open()
Variable_Object.Close()

```

Ex:

```

Dim cn As New OleDb.OleDbConnection
cn.ConnectionString="Provider=Microsoft.Jet.OleDb.4.0;Data Source=c:\levela.mdb"
cn.Open()
cn.Close()

```

2-របៀបប្រើ Connection ទៅកាន់ Ms SQL Server:

Syntax:

```

Keyword Variable_object As New OleDb.OleDbConnection
Variable_object.ConnectionString="Provider=SQLOLEDB;Data Source=ServerName or
IP Address; Initial Catalog=databasename;User Id=sqlserveruser;Password=sqlserverpassword"
Variable_object.Open()
Variable_Object.Close()

```

*Note: បើយើងចង់ connect ទៅកាន់នៅក្នុងម៉ាស៊ីនដែលសរសេរកូដផ្ទាល់យើងអាចធ្វើដូចខាងក្រោម:

```

Variable_object.ConnectionString="Provider=SQLOLEDB;Data Source=ServerName or
IP Address; Initial Catalog=databasename;Integrated Security=SSPI"

```




Ex:

```
Dim cn As New OleDb.OleDbConnection
cn.ConnectionString="Provider=SQLOLEDB;Data Source=127.0.0.1;Initial Catalog=ubb;User
Id=sa;Password=123"
cn.Open()
cn.Close()
```

enteritc.com

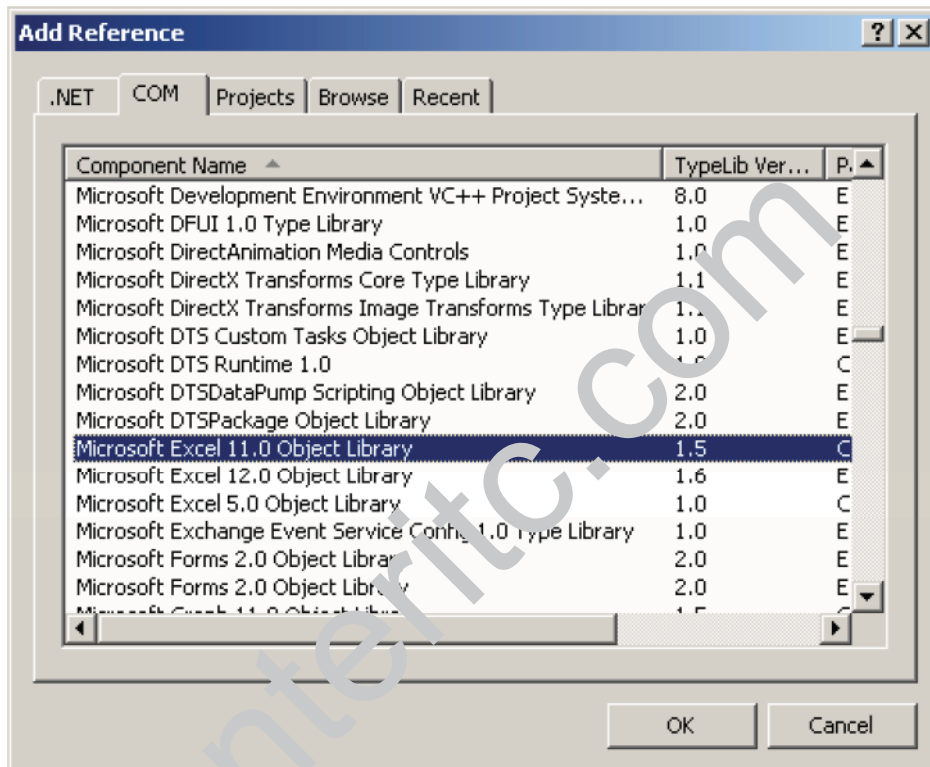
បេរៀនទី ១០

Report

I- Microsoft Exel:

*Add Reference:

-Click Project→Add Reference→Com→Microsoft Excel x.xx Object Library



-Ok

Syntax:

Imports Microsoft.Office.Interop

Keyword Variable As New Excell.Application

Keyword Variable As Excel.WorkBook

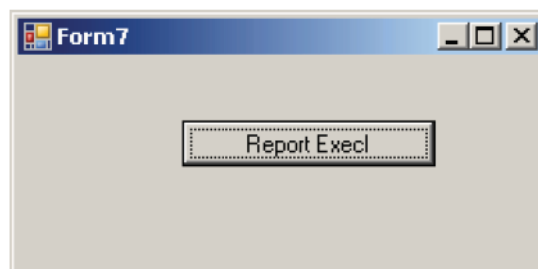
Keyword Variable As Excel.WorkSheet

Variable_Workbook = VariableApplication.Workbooks.Add

Note: បើយើងត្រូវការប្រើប្រាស់ template ដែលយើងបានបង្កើតនៅក្នុង excel ដែលមានស្រាប់ ។

wkbook = exapp.Workbooks.Open("D:\UBB\VB.netBook\test\test.xls")

Ex:



```

Imports Microsoft.Office.Interop
Imports System.Data.OleDb
Public Class Form7
    Dim cn As New OleDbConnection
    Dim cm As New OleDbCommand
    Dim exapp As New Excel.Application
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim wkbook As Excel.Workbook
        Dim wksheet As Excel.Worksheet
        Dim dr As OleDbDataReader
        wkbook = exapp.Workbooks.Add
        wksheet = wkbook.Worksheets("Sheet1")
        wksheet.PageSetup.PaperSize = Excel.XlPaperSize.xlPaperA4
        wksheet.PageSetup.LeftMargin = 0.5
        wksheet.PageSetup.RightMargin = 0.5
        wksheet.PageSetup.TopMargin = 0.5
        wksheet.PageSetup.BottomMargin = 0.5
        wksheet.PageSetup.CenterHorizontally = True
        wksheet.Cells("1", "C") = "Tariff"
        wksheet.Cells("1", "C").font.size = 16
        wksheet.Cells("1", "C").font.bold = True
        wksheet.Cells("1", "C").interior.colorindex = 3
        wksheet.Cells("3", "B") = "Code"
        wksheet.Cells("3", "B").borders.color = RGB(0, 0, 0)
        wksheet.Cells("3", "C") = "Item"
        wksheet.Cells("3", "C").borders.color = RGB(0, 0, 0)
        wksheet.Cells("3", "D") = "Unit"
        wksheet.Cells("3", "D").borders.color = RGB(0, 0, 0)
        wksheet.Cells("3", "E") = "Unit Price"
        wksheet.Cells("3", "E").borders.color = RGB(0, 0, 0)
        cm.CommandText = "Select * from tariff"
        dr = cm.ExecuteReader
        Dim j As Integer = 3
        Do While dr.Read
            j += 1
            wksheet.Cells(j, "B") = dr(0)
            wksheet.Cells(j, "E").borders.color = RGB(0, 0, 0)
            wksheet.Cells(j, "C") = dr(1)
            wksheet.Cells(j, "C").borders.color = RGB(0, 0, 0)
            wksheet.Cells(j, "D") = dr(2)
            wksheet.Cells(j, "D").borders.color = RGB(0, 0, 0)
            wksheet.Cells(j, "E") = dr(3)
            wksheet.Cells(j, "E").borders.color = RGB(0, 0, 0)
        Loop
        dr.Close()
        exapp.Visible = True
        wkbook.PrintPreview()
        wkbook.Close(False)
        exapp.Visible = False
    End Sub

    Private Sub Form7_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        cn.ConnectionString = "Provider=Microsoft.Jet.OleDb.4.0;Data
Source=D:\UBB\VB.netBook\test\levela.mdb"
        cn.Open()
        cm.Connection = cn
        cm.CommandType = CommandType.Text
    End Sub
End Class

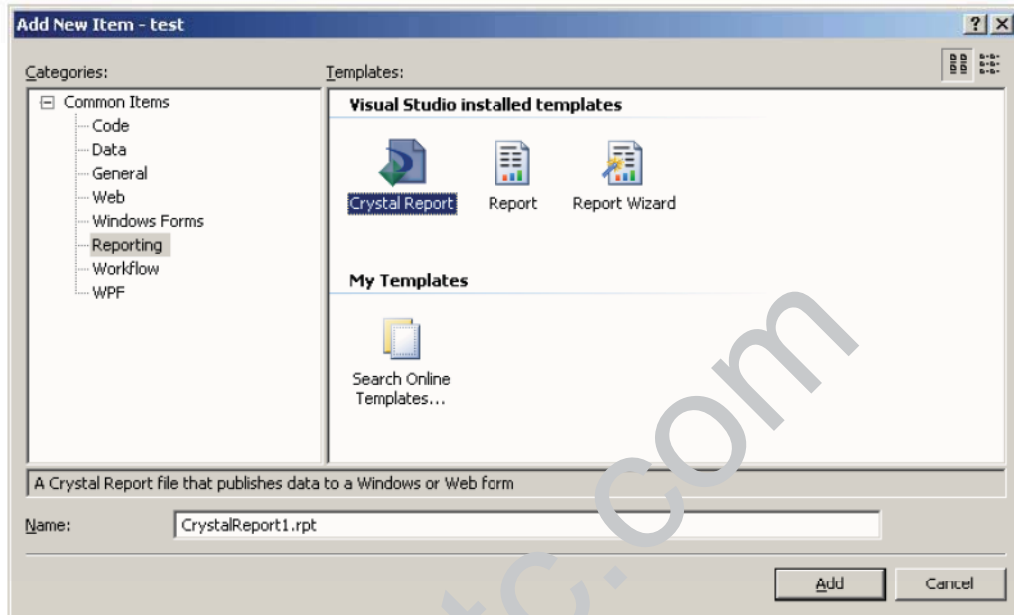
```

II- Crystal Report:

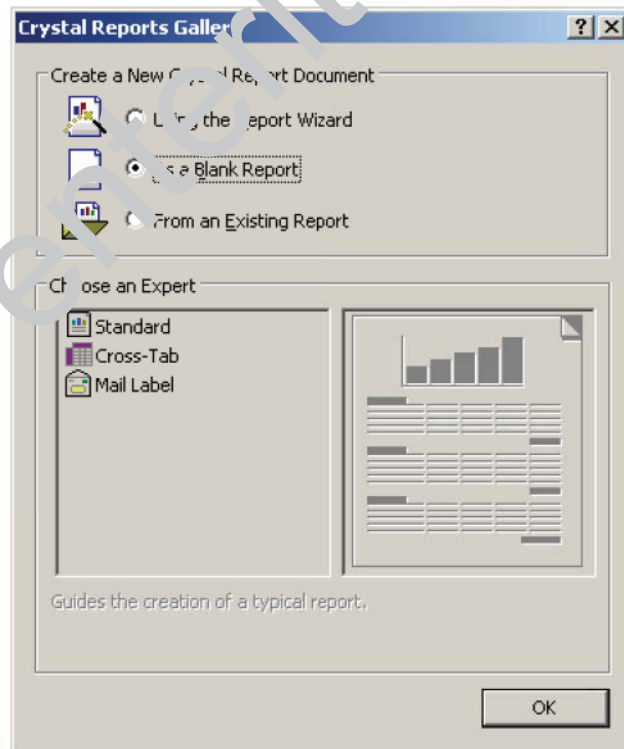
គឺជា application មួយដែលបង្កើតដោយក្រុមហ៊ុន Microsoft ក្នុងគោលបំណងសំរាប់ធ្វើរបាយការណ៍ផ្សេងៗ ។ វាជាកម្មវិធីមួយដែលមានប្រជាប្រិយជាងគេក្នុងការធ្វើរបាយការណ៍ ។

1- បង្កើត Report ជាមួយ Crystal Report ក្នុងកម្មវិធី Visual Studio 2005

-Click Project Menu → Add New Item



-Add → Accept



-Ok

+Report Header Section: មានតួនាទីសំរាប់បង្ហាញតួអក្សរ ឬរូបភាពតែមួយទំព័រដំបូងនៅផ្នែក header ។

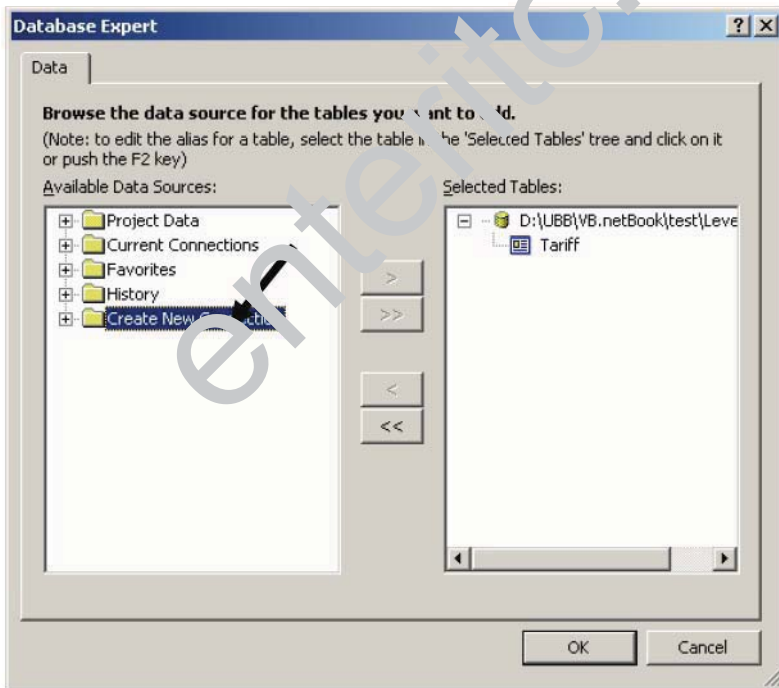
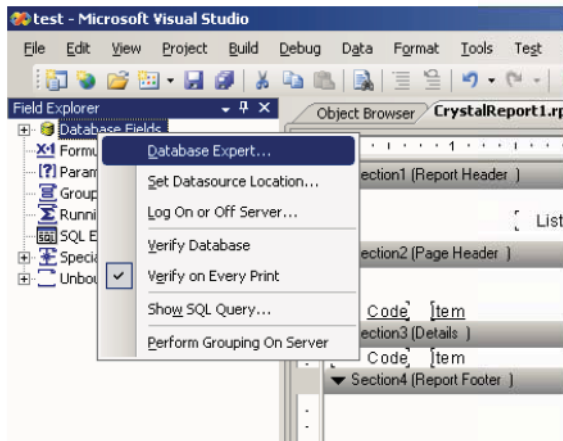
+Page Header Section: លោកអ្នកអាចដាក់តួអក្សរ ឬរូបភាពនៅត្រង់ចំណុចនេះ នោះពួកយើងនឹងបង្ហាញចេញគ្រប់ទំព័រទាំងអស់ ។

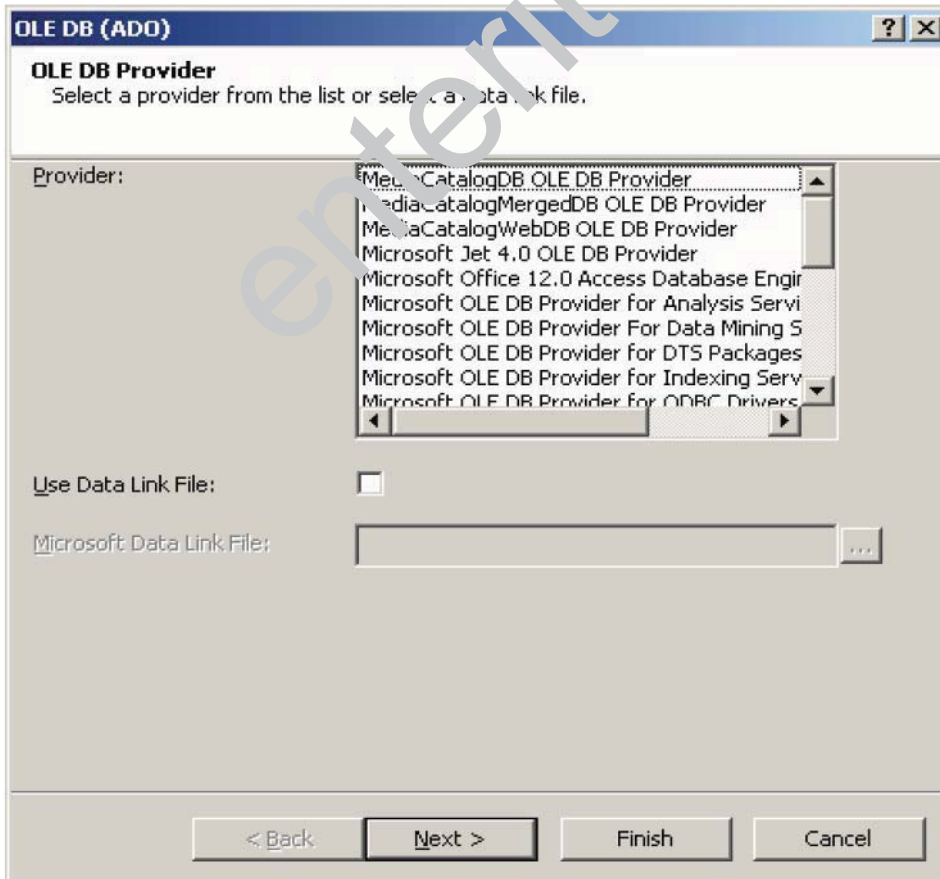
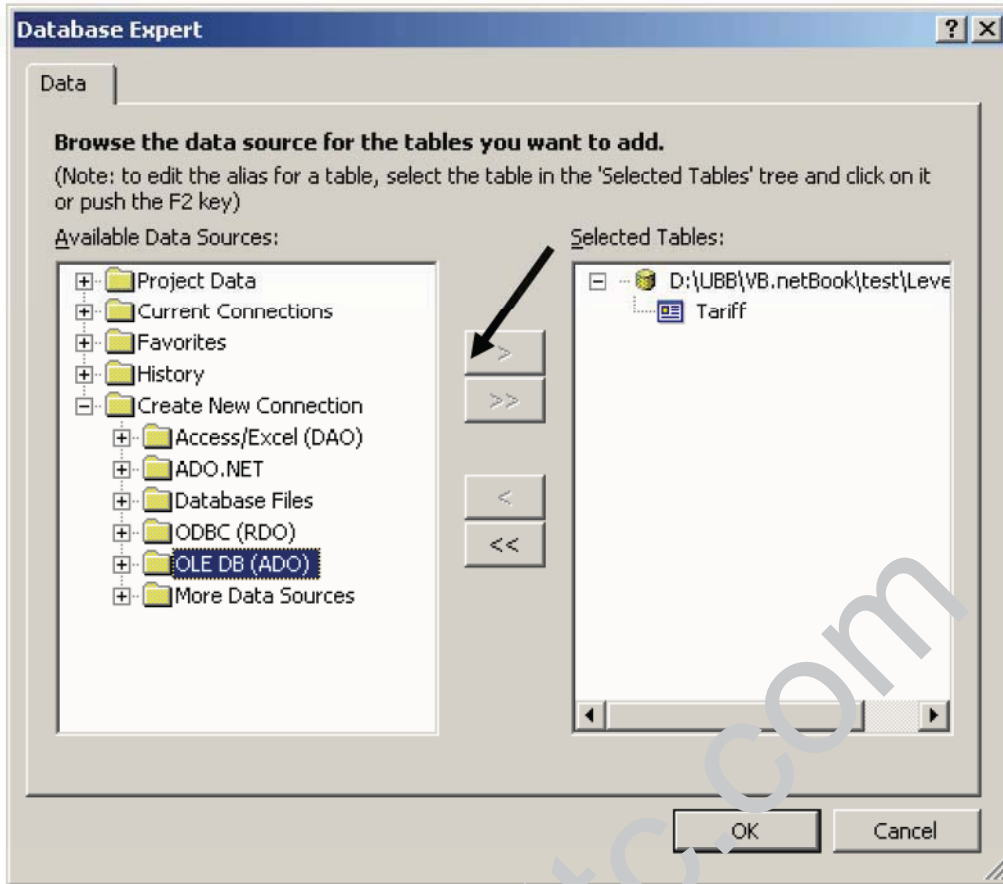
+Detail Section: គឺជាផ្នែកមួយដែលប្រើសំរាប់ទាញយកទិន្នន័យពី table ឬ query ។

+Report Footer: មានតួនាទីសំរាប់បង្ហាញតួអក្សរ ឬរូបភាពនៅទំព័រចុងក្រោយក្នុងផ្នែក footer ។

+Page Footer: មានតួនាទីសំរាប់បង្ហាញតួអក្សរ ឬរូបភាពនៅត្រង់ទំព័រនៅផ្នែក footer ។

2- ទាញយក Table ឬ Query មកដាក់លើ Report

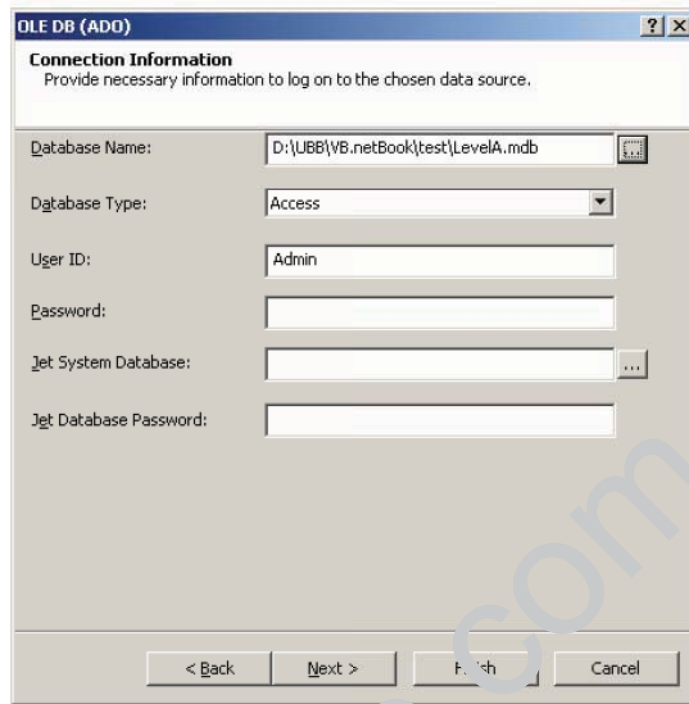




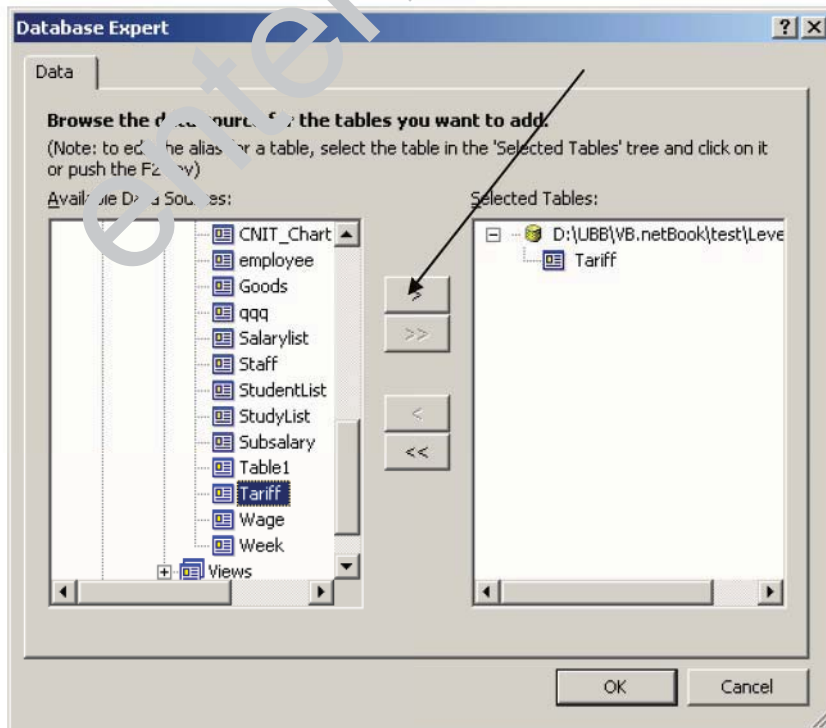
-ជ្រើសរើស Provider ណាកម្មកម្រិតបាន

Ex: Microsoft Jet 4.0 OLEDB Provider

-Click Next



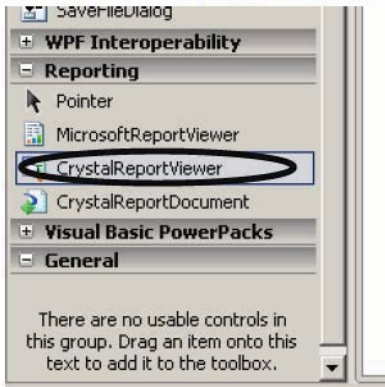
-Next→Finish



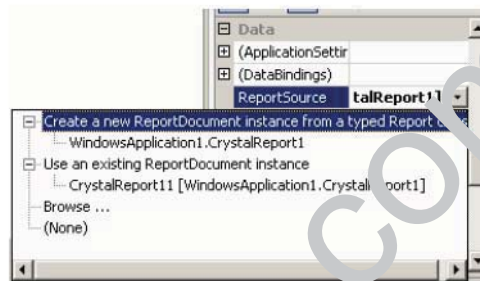
-Ok

3- របៀបដាក់ Report នៅក្នុង Form:

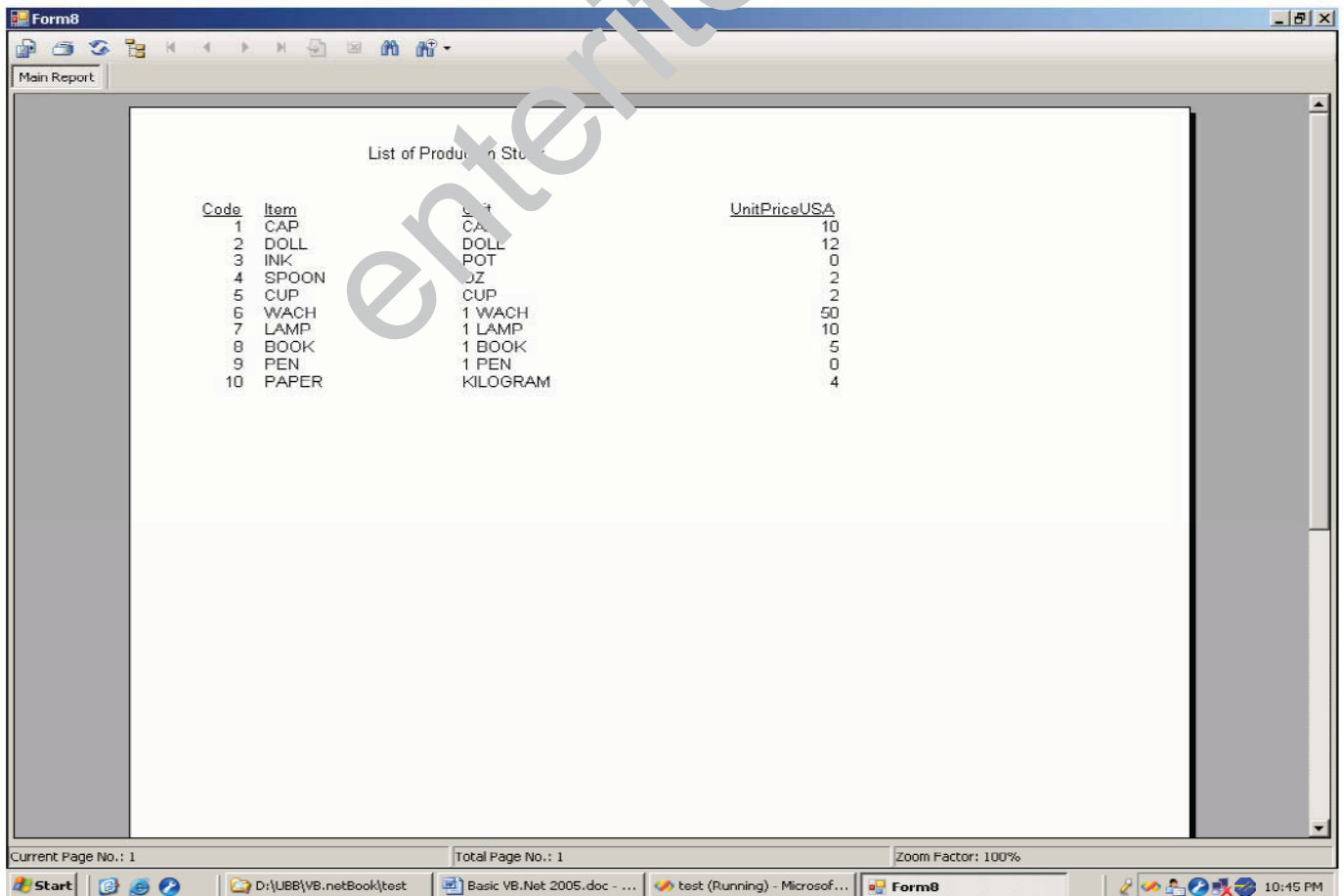
- ក្លឹប Crystal Report Viewer នៅលើ Form



- ជ្រើសរើស Report Source



Result:



3- ប្រើប្រាស់ទិន្នន័យលើ Report ពី Form:



```

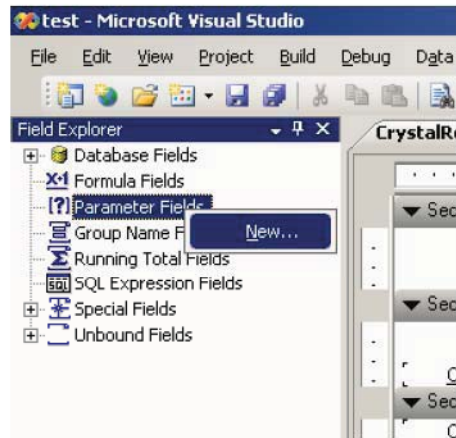
Public Class Form9
    Private Sub btnview_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnview.Click
        Form8.Show()
    End Sub
End Class

Public Class Form8
    Dim rpt_product As New CrystalReport1
    Private Sub CrystalReportViewer1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CrystalReportViewer1.Load
        If Form9.txtcode.Text <> "" Then
            rpt_product.RecordSelectionFormula = "{tariff code}=" & Form9.txtcode.Text
            Me.CrystalReportViewer1.ReportSource = rpt_product
        Else
            Me.CrystalReportViewer1.ReportSource = rpt_product
        End If
    End Sub
End Class

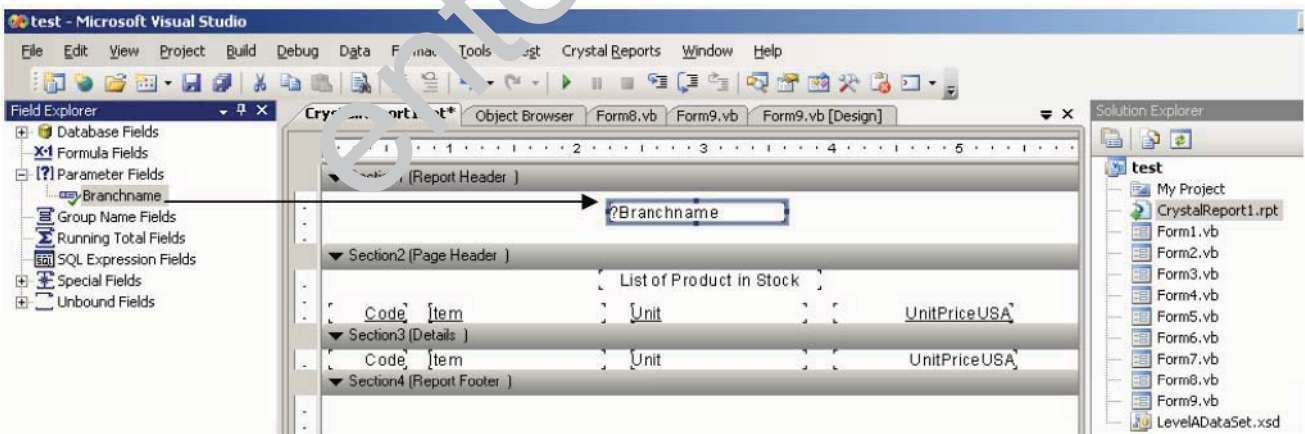
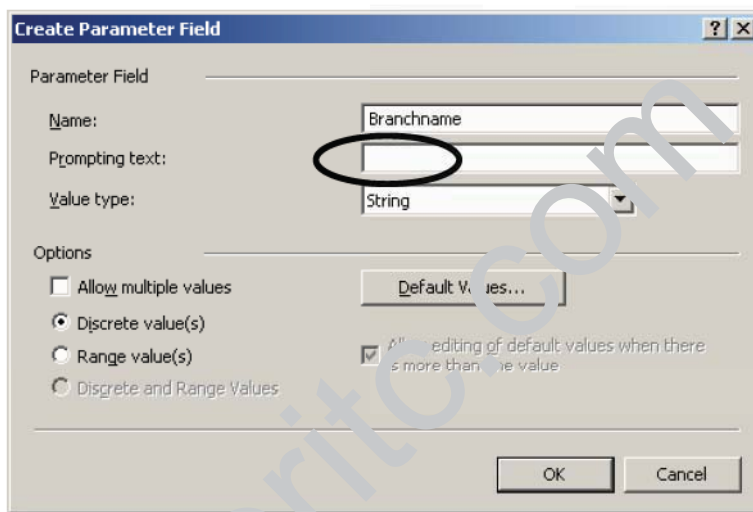
or
Imports System.Data.OleDb
Imports System.Data
Public Class Form8
    Dim cn As New OleDbConnection
    Private Sub CrystalReportViewer1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CrystalReportViewer1.Load
        cn.ConnectionString = "Provider=Microsoft.Jet.OleDB.4.0;Data Source=D:\UBB\VB.netBook\test\levela.mdb"
        cn.Open()
        Dim adpt As New OleDbDataAdapter("Select * from tariff where code=" & Form9.txtcode.Text, cn)
        Dim ds As New DataSet
        adpt.Fill(ds)
        Dim report As New CrystalReport1
        report.SetDataSource(ds.Tables(0))
        CrystalReportViewer1.ReportSource = report
        CrystalReportViewer1.Show()
    End Sub
End Class

```

4- របៀបបង្កើត Parameter Field:



-ដាក់ឈ្មោះអថេរ



Ex:

```

Public Class Form8
    Dim rpt_product As New CrystalReport1
    Private Sub CrystalReportViewer1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CrystalReportViewer1.Load
        If Form9.txtcode.Text <> "" Then
            rpt_product.RecordSelectionFormula = "{tariff.code}=" &
            Form9.txtcode.Text
            Me.CrystalReportViewer1.ReportSource = rpt_product
        Else
            Me.CrystalReportViewer1.ReportSource = rpt_product
        End If
    End Sub
End Class

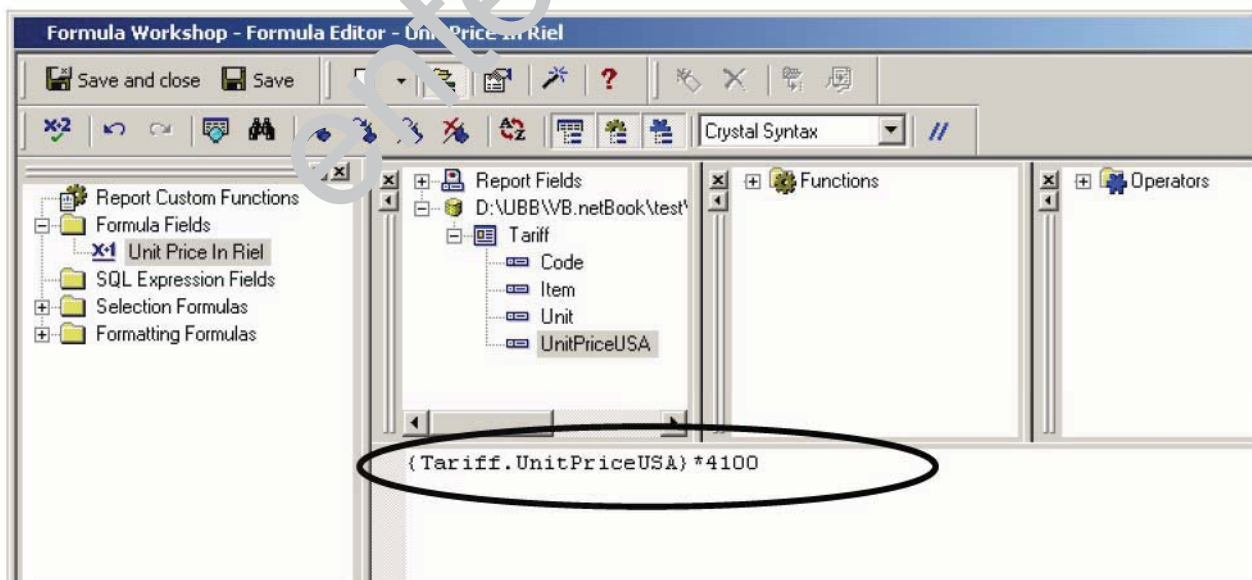
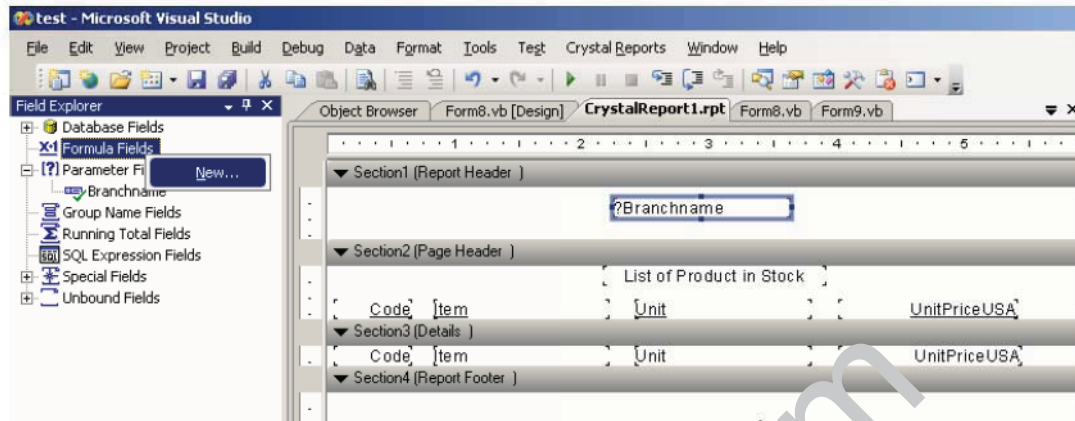
```

```

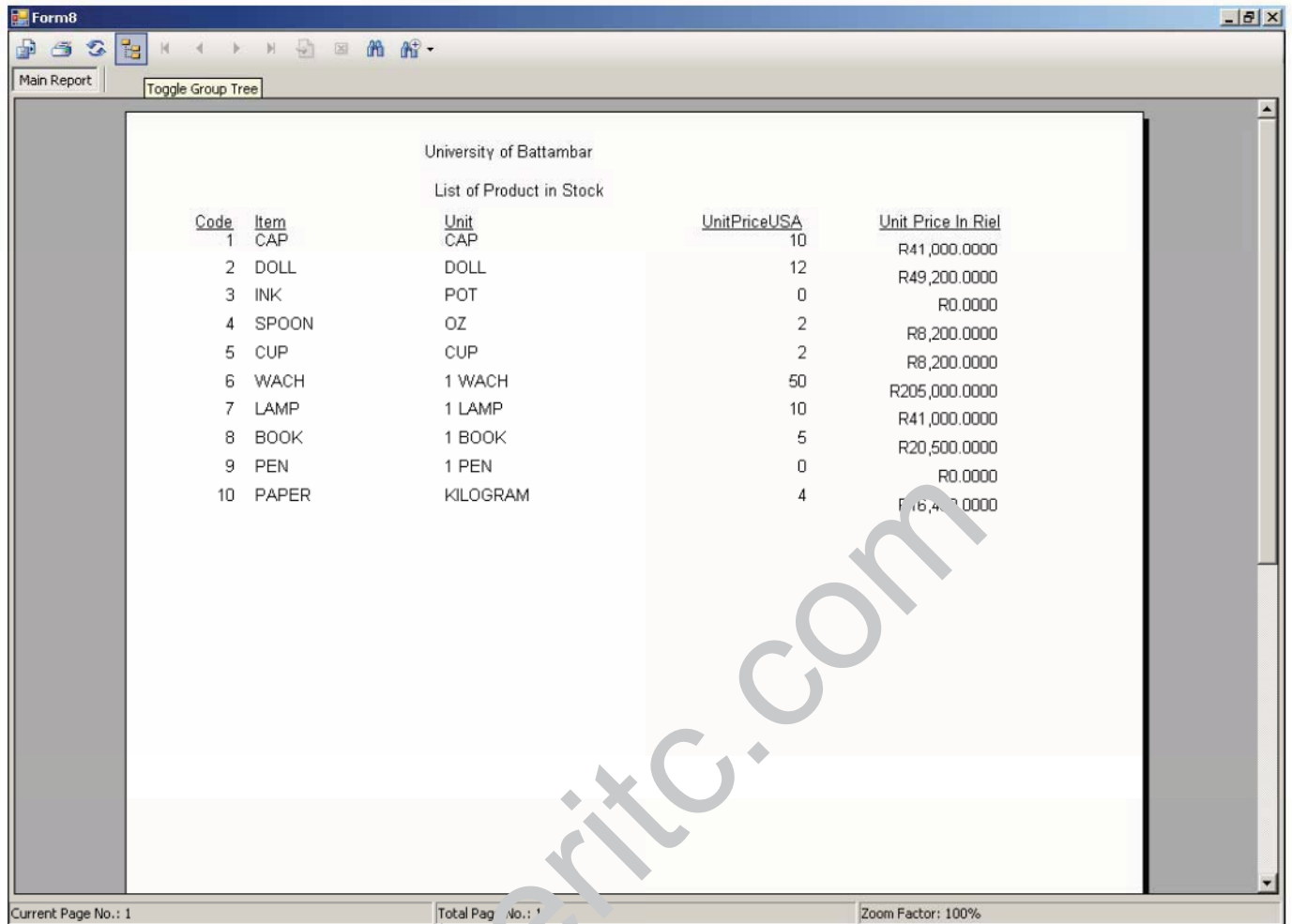
End If
Dim x As New CrystalDecisions.Shared.ParameterDiscreteValue
x.Value = "University of Battambang"
CrystalReportViewer1.ParameterFieldInfo("branchname").CurrentValues.Add(x)
End Sub
End Class

```

5- កម្រិតប្រើ Formula Field:

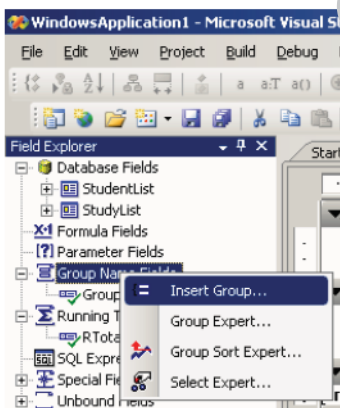


Ex:



6- ក្រុមរាយការណ៍ Grouping Report:

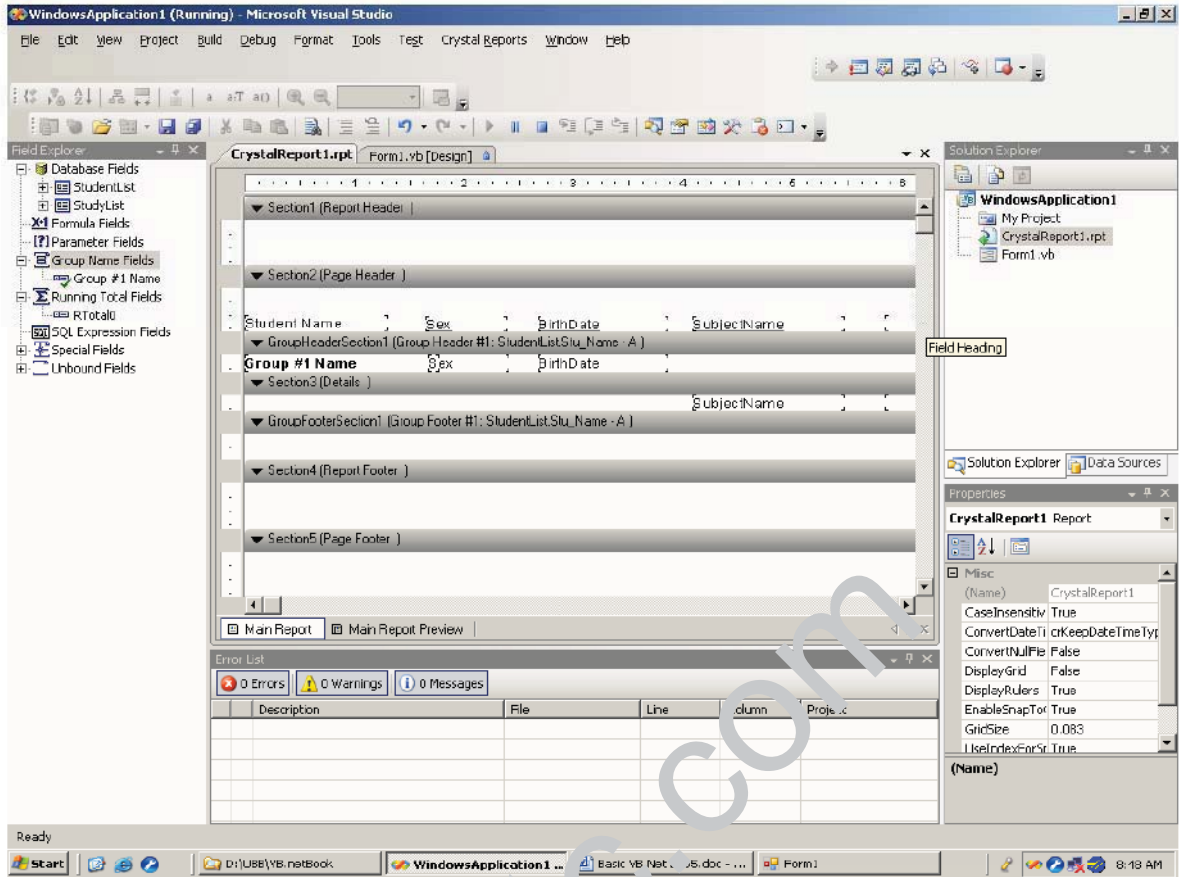
សំរាប់បង្កើត report ដោយបែងចែកវាជាក្រុមៗតាមផ្នែកមួយៗ



ជ្រើសរើស Field ដែល
ត្រូវបែងក្រុមដាក់ផ្ទុំ



-Click Ok



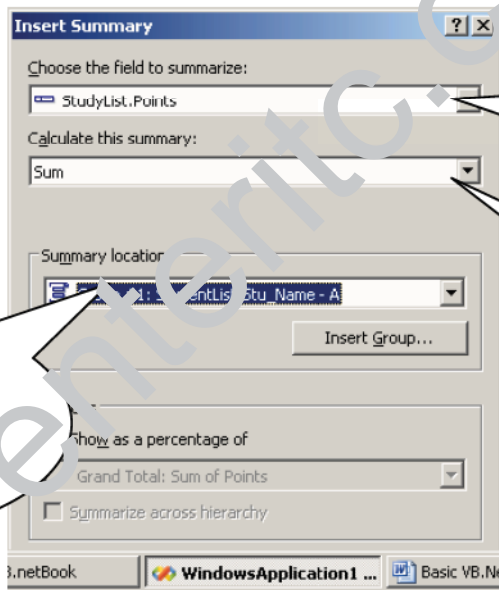
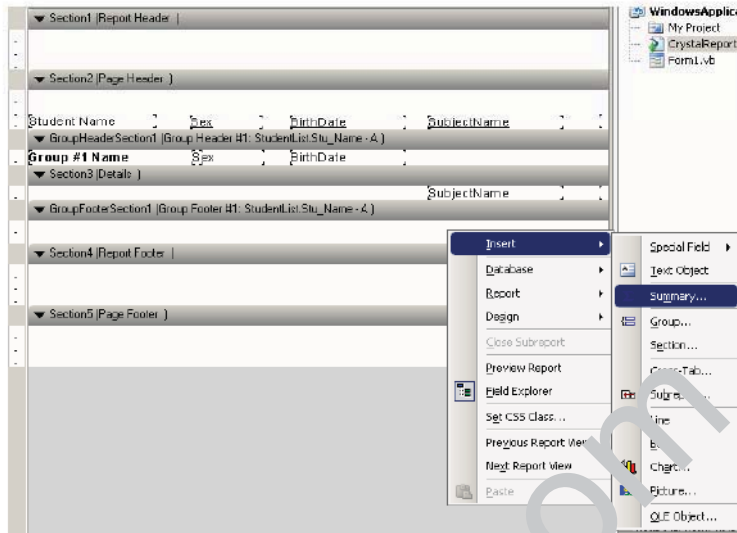
Ex:

Student Name	Sex	Birth Date	Subject Name	Points
ANG KIMNAY	M	10/24/1980 12:00:00	Statistic	8.50
			Visual Basic	10.00
			MIS	9.00
			Internet Programming	9.00
			Networking	9.50
KIM KOSAL	F	2/21/1978 12:00:00	Statistic	8.00
			Networking	4.50
			Internet Programming	5.50
			MIS	6.50
			Visual Basic	7.50
NAK POTHEA	F	12/21/1979 12:00:00	Visual Basic	5.00
			MIS	4.50
			Internet Programming	4.00
			Networking	5.00
			Statistic	4.50
Ong Khoeun	M	6/17/1980 12:00:00	Visual Basic	5.50
			MIS	6.00
			Internet Programming	7.00
			Networking	8.50
			Statistic	5.50
POK PHANY	M	10/6/1979 12:00:00	Networking	6.50
			MIS	9.00
			Visual Basic	7.50
			Statistic	3.50
			Internet Programming	8.00
JY MENG	M	1/9/1977 12:00:00A	Statistic	5.00
			Visual Basic	7.50
			MIS	9.00
			Internet Programming	6.50
			Networking	4.00

7- បង្កើតគណនា Sub Total:

គឺជាការបូកសរុបតាមផ្នែកមួយៗ

-Right Click in Group Footer



ជ្រើសរើស field ដែលត្រូវ
គណនា Subtotal

ជ្រើសរើស function
ដែលត្រូវត្រូវការ

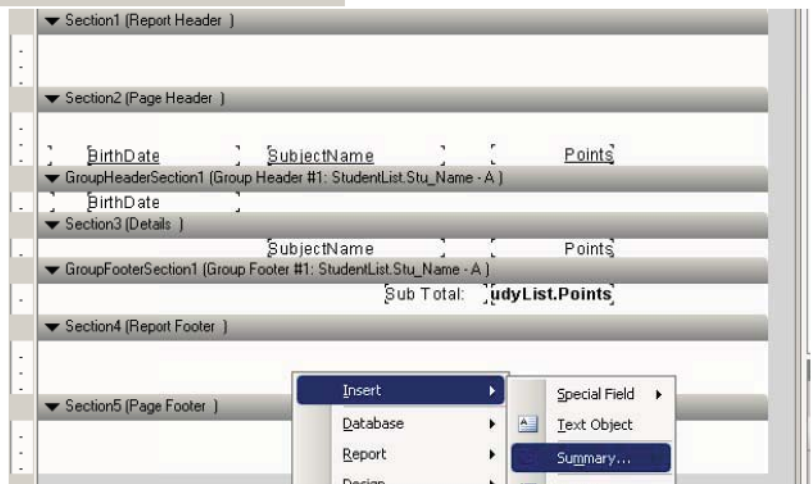
ជ្រើសរើសឈ្មោះ Group
ដែលត្រូវធ្វើប្រមាណវិធី

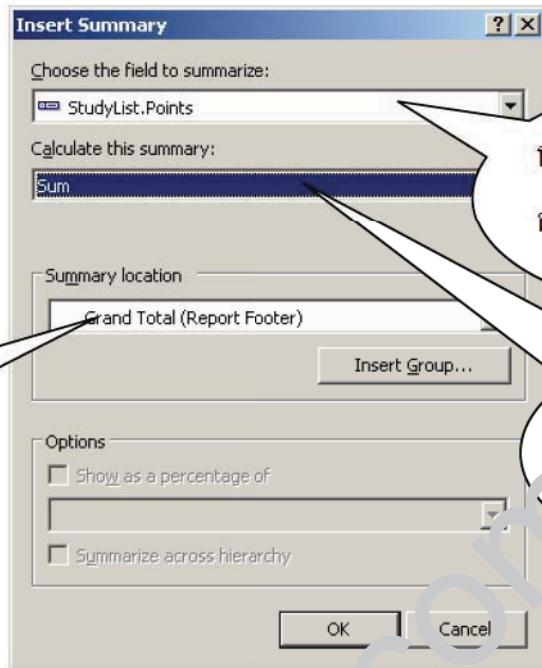
-Ok

7- បង្កើតគណនា Grand Total:

គឺជាការបូកសរុបរួមគ្រប់ផ្នែកទាំងអស់

-Right Click in Report Footer





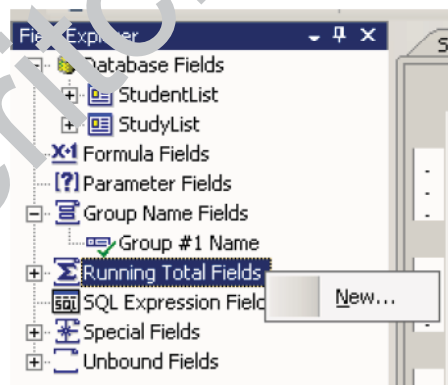
ជ្រើសរើស field ដែលត្រូវ
គណនា Grand Total

យកពាក្យ Grand Total

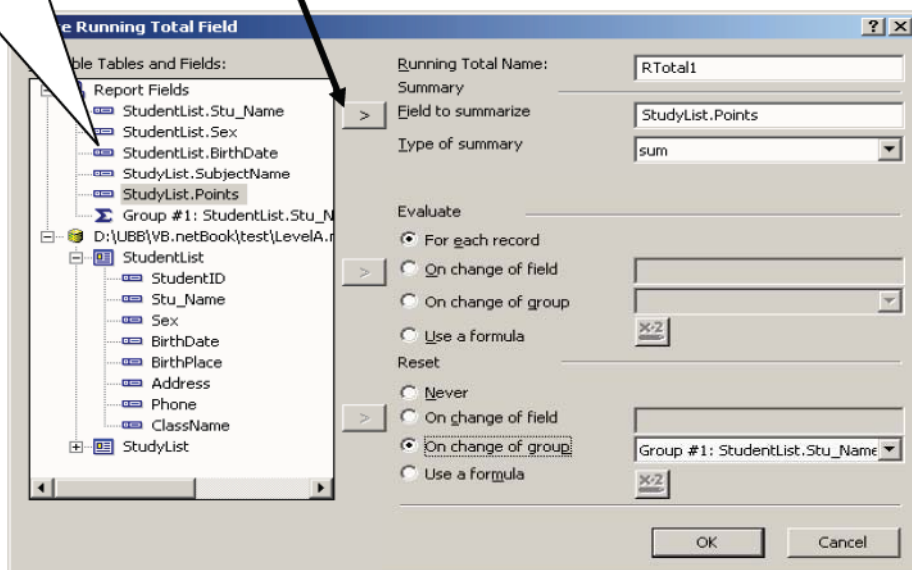
ជ្រើសរើស function
ដែលត្រូវត្រូវការ

8- របៀបបង្កើត Running Total:

សំរាប់ដាក់លេខរៀងដោយស្វ័យប្រវត្តិ



ជ្រើសរើស field ដែល
ត្រូវដាក់លេខរៀង



Ex:

Student Name	Sex	BirthDate	SubjectName	Points
ANG KIMHAY	M	12/24/1980 12:00:00	1 Statistic	8.50
			2 Visual Basic	10.00
			3 MIS	9.00
			4 Internet Programming	9.00
			5 Networking	9.50
Sub Total:				46.00
KIM KOSAL	F	2/21/1978 12:00:00	1 Statistic	6.00
			2 Networking	4.50
			3 Internet Programming	5.50
			4 MIS	6.50
			5 Visual Basic	7.50
Sub Total:				30.00
NAK POTHEA	F	12/12/1979 12:00:00	1 Visual Basic	5.00
			2 MIS	4.50
			3 Internet Programming	4.00
			4 Networking	5.00
			5 Statistic	4.50
Sub Total:				23.00
Ong Khoeun	M	6/17/1980 12:00:00	1 Visual Basic	5.50
			2 MIS	7.00
			3 Internet Programming	8.50
			4 Networking	5.50
			5 Statistic	5.50
Sub Total:				32.00
POK PHANY	M	10/6/1979 12:00:00	1 Networking	6.50
			2 MIS	9.00
			3 Visual Basic	7.50
			4 Statistic	3.50
			5 Internet Programming	8.00
Sub Total:				34.50
UY MENG	M	1/9/1977 12:00:00A	1 Statistic	5.00
			2 Visual Basic	7.50
			3 MIS	9.00
			4 Internet Programming	6.50
			5 Networking	4.00
Sub Total:				32.00
Grand Total				197.80

Current Page No.: 1

Total Page No.: 1

Zoom Factor: 83%

