Tutorial Sheet 6/Week 7

IMS1906: Business Software Fundamentals

Tutorial exercises
Week 7: Debugging Exercises

Aims
• To gain experience and familiarity with VB.NET IDE Debugger
• To understand the different types of programming errors that a programmer may encounter
• To understand what is meant by data validation
• To manage a VB.NET application

Task 1

a) Data validation guarantees to your application that every data value is correct and accurate. There are several types of data validation, explain what is meant by the following.
  • Data type validation
  • Range checking.

Give examples to illustrate your answer.
  b) How would you check that a variable is numeric?
  c) How would you check that your data lies between 0 and 1000
  d) How would you check that a string is alphabetic?

Task 2  Understanding programming errors

Consider the following extract of VB.NET code, which contains multiple errors:

```vbnet
1 Dim sngDeposits As Single, sngWithdrawals As Single
2 Dim sngBalance As Single, intMonth As Integer
3
4    intMonth = 1
5    sngBalance = 0
7
8    Do While (intMonth < 12)    ' Read 12 months worth of data
9        sngDeposits = txtDeposit.Text
10       sngWithdrawals = txtWithdrawal.Text
11       sngBalance = sngBalance + sngDeposits - sngWithdrawals
12       lstResultList.Items.Add( "Balance for month " & Month & " is " & _
13       sngBalance.ToString( "C")
14   End Do
15
16   lblFinalBalance.Text = sngBalance
```

Please contact your tutor if you require assistance with these exercises. Tutor email address and consultation times are available on the subject’s Web page under Staff.
a) For each error:
   i. state which line(s) it occurs at and explain what is causing the problem to occur.
   ii. state what type of error it is (logic, syntax or run-time), and
   iii. explain how the error might have been detected if you were the person who had typed this code.

b) Re-write the lines that should be modified to correct the program.

Task 3     To debug the VB.NET programs.
Download the following VB.NET Programs from the subjects website, under the tutorial link. These VB.NET programs have got some bugs in them:

- debug-task-1.zip
- debug-task-2.zip

Use the debugging tools available in the Visual Studio .NET and the techniques presented in this week's lecture materials to debug each of the programs. Set breakpoints at suitable places, establish watches for appropriate variables and expressions, then step through the code examining the behaviour of the program. **Do not just change the code** - this is an exercise in practicing debugging techniques.

Questions:
- What is the program supposed to be doing?
- What types of errors do you find (are they logic, run-time or syntax errors) in each program? Where do they occur, and why?
- What changes do you need to make to the code, to make the program work correctly?

The actual VB.NET code is provided in the Appendix for you to browse and insert comments such as where you inserted break points and what variable and expressions were watched.

Your tutor *may* return your mark to TEST 1 and discuss any concern you may have.
Appendix 1 – Debug-Task-1

Private Sub cmdCount_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdCount.Click

' This program counts from the lower of two numbers
' through to the higher of the two.

Dim intStart As Integer, intFinish As Integer
Dim intCounter As Integer

' Determine the values the user has entered
intStart = Convert.ToInt32(Val(txtFirstNum.Text))
intFinish = Convert.ToInt32(Val(txtSecondNum.Text))

' Ensure we start with the lower number
If intStart >= intFinish Then
    intStart = Convert.ToInt32(Val(txtSecondNum.Text))
    intFinish = Convert.ToInt32(Val(txtFirstNum.Text))
End If

txtOutput.Text = ""

' Go from the lower to the higher number
For intCounter = intStart To intFinish
    ' Display the current value of intCounter
    txtOutput.Text &= intCounter & " "
    ' Increment for next iteration.
    intCounter = intCounter + 1
Next intCounter

End Sub
End Class
Exercise 2 Debug-Task-2

Private Sub cmdConvert_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdConvert.Click
    Dim intMark As Integer

    ' Obtain the mark
    intMark = Convert.ToInt32(txtAsgMark.Text)
    txtOutput.Text = "Processing Mark: " & intMark

    ' Determine the standard Monash Grade for the Mark.
    If intMark > 100 Or intMark < 0 Then
        txtOutput.Text = txtOutput.Text & (" Mark not valid")
    Else
        If intMark > 80 Then
            txtOutput.Text = txtOutput.Text & (" You get HD")
        End If
        ElseIf intMark > 70 Then
            txtOutput.Text = txtOutput.Text & (" You get a D")
        ElseIf intMark > 60 Then
            txtOutput.Text = txtOutput.Text & (" You get a C")
        End If
        If intMark > 50 Then
            txtOutput.Text = txtOutput.Text & (" You only got a P")
        ElseIf intMark < 50 Then
            txtOutput.Text = txtOutput.Text & (" Try harder next time")
        End If
    End If
End Sub
End Class