A201 Object Oriented Programming with Visual Basic .Net

By:

Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend
What do we need to learn in order to write computer programs?

- Fundamental programming constructs:
  - Variables,
  - Arithmetic operators,
  - Input and output
  - Conditionals,
  - Loops,
  - Procedures and functions,
  - Arrays (Multi-Dimensional Arrays),
  - Structures, classes and objects,
  - Files
What is a File?

- Collection of bits stored in a **persistent** secondary storage device.
What is a File?

☐ The bits in the file may represent:

☐ Text
☐ Numbers
☐ Image
☐ Audio
☐ Video
☐ Etc.

(c) Copyright 2007, H. Hakimzadeh
What is Persistence?

☐ State or memory that survives failures.

☐ Data's ability to survive beyond the execution of a program.
Persistent vs. Volatile Memory:

- **Persistent memory** contrasts with **volatile memory** which is reset at system restart.
File Organizations:

- Sequential
- Random Access
- Index Sequential
Sequential File Organization:

- Sequential Access
- Consecutively Ordered Records.
- Records can have varying length.
- Records are read and written sequentially.
- Simple organization. (simple to program)
- On average n/2 comparisons are needed to find a record.
- To update or delete a record, the file must be re-written.
- Searching, deleting and updating operations are very slow when number of records (n) becomes large.

(c) Copyright 2007, H. Hakimzadeh
Random File Organization:

- Direct Access (Fast).
- Fixed record size.
- Location of the records are computed by using their unique key (identifier)
  - Record_Location = Record_ID * Record_length
- Logical ordering of records may or may not correspond to their physical sequence.
- Insertion, deletion and searching of records can be done at random. (very fast)
- Random access files can be updated in place.
Index-Sequential File Organization:

Provides both Sequential and Direct Access

Combines sequential access and ordering with random access capabilities.

Contains two parts:
1) A sequential file:
   Maintains the actual data.
   Sometimes ordered on a key.
   Variable sized records.

2) An Index:
   A linear or hierarchical index structure:
   (Key, Block #) or
   (Key, Absolute file location)
Files in VB .Net:

- In .Net framework, I/O centers around the stream.

- Stream is designed to transfer a series of bytes from one location to another.

- Streams are objects that have properties and methods. These objects are found in the System.IO namespace.

  * Imports system.io

- File handling projects must contain an “Imports” statement. (before the statement declaring the form class)
Steps for Writing to a File:

1. Get some input from the user (from a text box or from the console).

2. Declare a new StreamWriter object. (Also declares the name of the data file)

3. Use StreamWriter's WriteLine() method. (Copies the data to a buffer in memory)

4. Call StreamWriter's Close() method. (Transfers the data from buffer to the file and releases system resources used by the stream)
Opening a File for Writing:

Open for Writing:
- Declaring a new **StreamWriter** object opens the file
- If the file **does not exist**, a new one is **CREATED**
- If the file **already exist**, it will be **ERASED**.

Syntax:
- Dim ObjectName As New StreamWriter("FileName")
- Dim ObjectName As New StreamWriter("FileName", BooleanAppend)

Example:
- Dim datPhone As New StreamWriter("Phone.txt")
- Dim datNames As New StreamWriter("C:\MyFiles\Names.txt")
- Dim datLogFile As New StreamWriter("Logfile.txt", True)

- Default location for files is the **“bin”** directory beneath the folder for the current project

(c) Copyright 2007, H. Hakimzadeh
Write() Method:

- Places items consecutively in the file with no delimiter (separator)

Syntax:
- FileObject.Write(DataToWrite)

Example:
- Dim OutputFile As New StreamWriter("data.txt")
- OutputFile.Write(txtName.Text)
- OutputFile.Write("A505")
- OutputFile.Write(decBalance.ToString)
Writeline() Method:

- Places a “newline” character, between items

Syntax:

- FileObject.WriteLine(DataToWrite)

Example:

- Dim OutputFile As New StreamWriter(“data.txt”)
- OutputFile.WriteLine(txtName.Text)
- OutputFile.WriteLine(“A505”)
- OutputFile.WriteLine(decBalance.ToString)
Close() Method:

- The close() method, finishes writing all data from the stream's buffer to the disk, closes the file and releases system resources associate with the file.

- **Syntax:**
  - FileObject.Close()

- **Example:**
  - `OutputFile.Close()`
Steps for Reading a File:

1. Declare a new **StreamReader** object (Opens the file)

2. Use StreamReader's **ReadLine()** method.

3. Typically a **loop** is used to retrieve multiple records.

4. Call StreamReader's **Close()** method
Opening a File for Reading:

Open for Reading:

- Declaring a new **StreamReader** object opens the file.
- If the file **does not exist**, an **ERROR** is generated.
- Declare the StreamReader object only in a procedure so that you can enclose it in a **Try/Catch** block for error handling.

**Syntax:**

- Dim FileObject As New StreamReader("FileName")

**Example:**

- **Try**
  - Dim InputFile As New StreamReader("Phone.txt")
- **Catch**
  - MessageBox.Show("File does not exist")
- **End Try**

---

(c) Copyright 2007, H. Hakimzadeh
Readline() Method:

- Each time it executes, it reads the next line of data (Just like reading from the keyboard)

- Syntax:
  - FileObject.ReadLine()

- Example:
  - Dim InputFile As New StreamReader("Phone.txt")
  - txtName.Text = InputFile.ReadLine()
  - lblPhone.Text = InputFile.ReadLine()
  - strName = InputFile.ReadLine()
Peek() Method:

- **Peek** looks at the next element without actually reading it. It is used to check if an END-of-FILE is reached.

- If we Peek beyond the last element of the file, a “-1” is returned.

- To be safe, code an IF structure to execute Peek and compare for < > -1 before reading.

**Syntax:**
- `ObjectName.peek()` ‘Return a -1 if at EOF

**Example:**
- `Dim inputFile As New StreamReader("Phone.txt")`
- `Do Until inputFile.Peek() = -1`
- `strOutput = inputFile.ReadLine()`
- `Console.WriteLine(strOutput)`
- `Loop`
Complete Example:

```vbnet
Imports System.IO

Module Module1

Sub Main()
    Dim FileName As String
    FileName = InputBox("Enter the file name:", "File Name")

    WriteToFile(FileName)

    ReadFromFile(FileName)

End Sub

Private Sub WriteToFile(ByVal FileName As String)
    'Read from the keyboard and write to a file...

End Sub

Private Sub ReadFromFile(ByVal FileName As String)
    'Read from a file and display it on the screen...

End Sub

End Module
```

Import the IO library

Write to file

Read from file

(c) Copyright 2007, H. Hakimzadeh
Complete Example:

```vbnet
Imports System.IO

Module Module1

Sub Main()
    Dim FileName As String
    FileName = InputBox("Enter the file name:", "File Name")
    WriteToFile(FileName)
    ReadFromFile(FileName)
End Sub

Private Sub WriteToFile(ByVal FileName As String)
    'Read from the keyboard and write to a file...
End Sub

Private Sub ReadFromFile(ByVal FileName As String)
    'Read from a file and display it on the screen...
End Sub

End Module
```

(c) Copyright 2007, H. Hakimzadeh
Complete Example:

```vbscript
Private Sub WriteToFile(ByVal FileName As String)
    FileName = FileName.Trim  ' Get rid of trailing spaces
    Dim A_NewLine As String
    If FileName <> "" Then
        Try
            Dim OutputFile As StreamWriter = New StreamWriter(FileName)

            Do
                A_NewLine = InputBox("Enter a new line (""q or Q"" to exit)", "Input a Line...")
                If A_NewLine.Trim.ToLower <> "q" Then
                    OutputFile.WriteLine(A_NewLine)
                End If
            Loop Until A_NewLine.Trim.ToLower = "q"

            OutputFile.Close()

            Catch
                Console.WriteLine("You must enter a valid file name to continue..")
            End Try
        Else
            MsgBox("Error you must enter a file name!")
        End If
    End If
End Sub
```

(c) Copyright 2007, H. Hakimzadeh
Complete Example:

```vbnet
Private Sub WriteToFile(ByVal FileName As String)
    FileName = FileName.Trim ' Get rid of trailing spaces
    Dim A_NewLine As String
    If FileName <> "" Then
        Try
            Dim OutputFile As StreamWriter = New StreamWriter(FileName)
            Do
                A_NewLine = InputBox("Enter a new line (""q or Q"" to exit), "Input a Line..."")
                If A_NewLine.ToLower <> "q" Then
                    OutputFile.WriteLine(A_NewLine)
                End If
            Loop Until A_NewLine.ToLower = "q"
            OutputFile.Close()
        Catch
            Console.WriteLine("You must enter a valid file name to continue..")
        End Try
    Else
        MsgBox("Error you must enter a file name!")
    End If
End Sub
```

(c) Copyright 2007, H. Hakimzadeh
Private Sub WriteToFile(ByVal FileName As String)
    FileName = FileName.Trim  ' Get rid of trailing spaces
    Dim A_NewLine As String
    If FileName <> "" Then
        Try
            Dim OutputFile As StreamWriter = New StreamWriter(FileName)
            Do
                A_NewLine = InputBox("Enter a new line (""q or Q"" to exit)", "Input a Line...")
                If A_NewLine.ToLower <> "q" Then
                    OutputFile.WriteLine(A_NewLine)
                End If
            Loop Until A_NewLine.ToLower = "q"
            OutputFile.Close()
        Catch
            Console.WriteLine("You must enter a valid file name to continue...")
        End Try
        Else
            MsgBox("Error you must enter a file name!")
        End If
    End If
End Sub
Complete Example:

Private Sub WriteToFile(ByVal FileName As String)
    FileName = FileName.Trim ' Get rid of trailing spaces
    Dim A_NewLine As String
    If FileName <> "" Then
        Try
            Dim OutputFile As StreamWriter = New StreamWriter(FileName)
            Do
                A_NewLine = InputBox("Enter a new line (""q or Q"" to exit)", "Input a Line...")
                If A_NewLine.ToLower <> "q" Then
                    OutputFile.WriteLine(A_NewLine)
                End If
            Loop Until A_NewLine.ToLower = "q"
            OutputFile.Close()
        Catch
            Console.WriteLine("You must enter a valid file name to continue..")
        End Try
    Else
        MsgBox("Error you must enter a file name!")
    End If
End Sub

(c) Copyright 2007, H. Hakimzadeh
Complete Example:

Private Sub WriteToFile(ByVal FileName As String)
    FileName = FileName.Trim    ' Get rid of trailing spaces
    Dim A_NewLine As String
    If FileName <> "" Then
        Try
            Dim OutputStream As StreamWriter = New StreamWriter(FileName)

            Do
                A_NewLine = InputBox("Enter a new line (""q or Q"" to exit)", "Input a Line...")
                If A_NewLine.ToLower <> "q" Then
                    OutputStream.WriteLine(A_NewLine)
                End If
            Loop Until A_NewLine.ToLower = "q"

            OutputStream.Close()
        Catch
            Console.WriteLine("You must enter a valid file name to continue..")
        End Try
    Else
        MessageBox("Error you must enter a file name!")
    End If
End Sub

(c) Copyright 2007, H. Hakimzadeh
Complete Example:

```vbscript
Private Sub ReadFromFile(ByVal FileName As String)
    FileName = FileName.Trim
    Dim strOutput As String
    If FileName <> "" Then
        Try
            Dim inputFile As StreamReader = New StreamReader(FileName)
            Console.WriteLine("Here is the contents of " & FileName & ":")

            Do Until inputFile.Peek() = -1
                strOutput = inputFile.ReadLine()
                Console.WriteLine(strOutput)
            Loop
        InputFile.Close()
        Catch
            Console.WriteLine("Unable to open " & FileName, "File Does Not Exist")
        End Try
    Else
        MsgBox("Error you must enter a file name!")
    End If

End Sub

End Module

(c) Copyright 2007, H. Hakimzadeh
Complete Example:

Private Sub ReadFromFile(ByVal FileName As String)
    FileName = FileName.Trim
    Dim strOutput As String
    If FileName <> "" Then
        Try
            Dim InputFile As StreamReader = New StreamReader(FileName)
            Console.WriteLine("Here is the contents of " & FileName & ":")
            Do Until InputFile.Peek() = -1
                strOutput = InputFile.ReadLine()
                Console.WriteLine(strOutput)
            Loop
            InputFile.Close()
        Catch
            Console.WriteLine("Unable to open " & FileName & ":")
        End Try
    Else
        MsgBox("Error: you must enter a file.
    End If
End Sub
End Module