Higher

Computing Science

Software Design and Development

Sequential File Handling

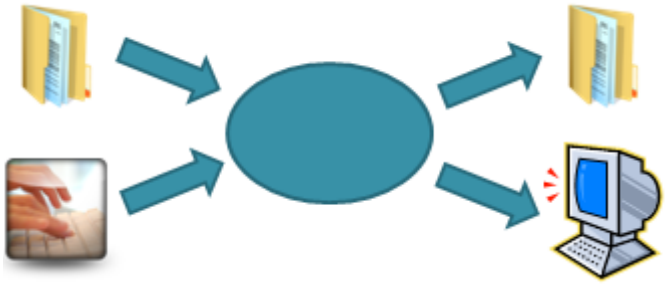
Name:_____

Contents

Sequential File Operations

File handling is an alternative method to using the keyboard and display for input and output.

The purpose of file operations is to enable information to be received directly from a text file or sent directly to a text file.



Sending data to a file allows the output to be permanently stored.

The stored data could then be read back into the program the next time it is executed.

Imagine a computer game kept a high scores table.

Without file handling, the high scores would only exist until the game was turned off. The next time you played, they would be gone.

File handling allows the scores to be saved by the

Input from Sequential Files

If we wanted to input a score from a text file we would write:

```
OPEN "mytextfile.txt"
    RECEIVE score FROM "mytextfile.txt"
CLOSE "mytextfile.txt"
```

Notice that the file has to be **opened** before it can be read and then **closed** again when the input is complete.

Output to Sequential Files

If we wanted to output a score to a text file we would write:

```
CREATE "mytextfile.txt"
OPEN "mytextfile.txt"
        SEND score TO "mytextfile.txt"
CLOSE "mytextfile.txt"
```

Sequential File Operations

Open:	Initialises a file to prepare it to be read from or written to
Create:	Establish a new file and give it a name
Read:	Copy data from a file and store it in memory (variable/array)
Write:	Copy data memory (variable/array) and place it in a file
Close:	Close a file

There are two methods of storing data in Sequential files.

- Single item per line
- Comma Separated Values (CSV)

Single Item per Line

Single item per line means that data would be stored one item above another as shown.

In this format, reading a whole line will give you a single piece of data.

```
Gemma
Wilson
14
Arran
James
Black
13
Bute
```

Comma Separated Values

CSV files store separate data items, each separated with a comma. Commonly, data related to a single person or thing would then be stored on separate lines.

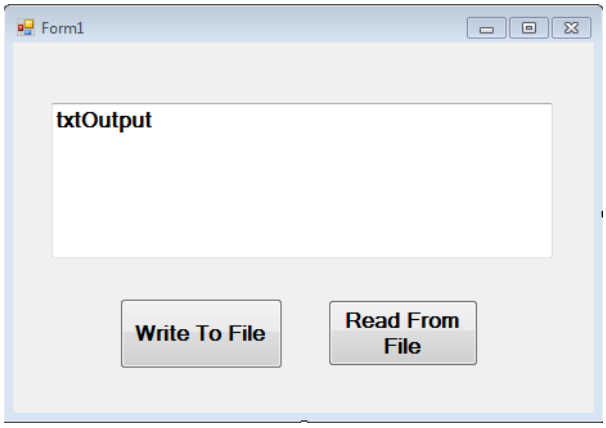
In this format, reading a whole line will give you a single piece of data.

```
Gemma,Wilson,14,Arran
James,Black,13,Bute
```

Single Item per Line - Worked Example 4a

This example will allow a single user's name and age to be appended to a sequential file each time the button is worked.

EOF is used to read in all the data currently in the file and display it on screen.




```
Public Class Form1


Private Sub btnWrite_Click(sender As Object, e As EventArgs) Handles btnWrite.Click

        Dim userName As String
        Dim userAge As Integer


        userName = InputBox("Please enter your name")
        userAge = InputBox("Please enter your age")

        FileOpen(1, "H:\userdetails.txt", OpenMode.Append)

        PrintLine(1, userName)
        PrintLine(1, userAge)

        FileClose(1)

End Sub
```

Append means the new data will be added to the end of the file

[Continued over page]

```
Private Sub btnRead_Click(sender As Object, e As EventArgs) Handles btnRead.Click

        Dim usernames(10) As String
        Dim userages(10) As Integer

        Dim counter As Integer


        FileOpen(1, "H:\userdetails.txt", OpenMode.Input)



        Do Until EOF(1)

            usernames(counter) = LineInput(1)
            userages(counter) = LineInput(1)


            counter = counter + 1

        Loop

        FileClose(1)




        For index = 0 To counter - 1

            txtOutput.AppendText(usernames(index) & vbTab & userages(index) &
vbNewLine)

        Next


End Sub
```

Notice the mode is now Input

These lines repeat until 'EOF' – End of File

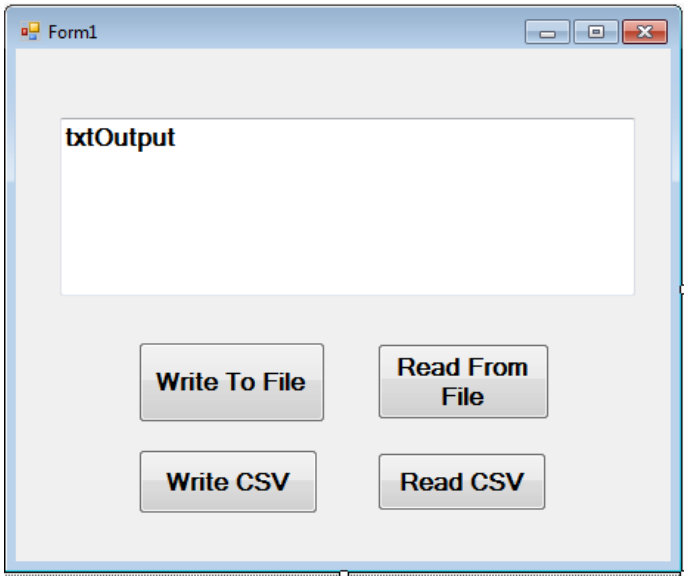
Comma Separated Value - Worked Example 4b

Do **not** start a new project.

Add two buttons to the previous example as shown.

This example shows how to create a CSV file using commas.

The data is then read in and separated using the Split command.




```
Private Sub btnWriteCSV_Click(sender As Object, e As EventArgs) Handles
btnWriteCSV.Click


        Dim userName As String
        Dim userAge As Integer


        userName = InputBox("Please enter your name")
        userAge = InputBox("Please enter your age")



        FileOpen(1, "H:\userdetails2.txt", OpenMode.Append)

        PrintLine(1, userName & "," & userAge)


        FileClose(1)


    End Sub
```

This line writes data as before but uses concatenation to insert a comma between the data.

[Continued over page]

```
Private Sub btnReadCSV_Click(sender As Object, e As EventArgs) Handles
btnReadCSV.Click


        Dim usernames(10) As String
        Dim userages(10) As Integer

        Dim separated(2) As String

        Dim counter As Integer


        FileOpen(1, "H:\userdetails2.txt", OpenMode.Input)

        Do Until EOF(1)

            separated = Split(LineInput(1), ",")

            usernames(counter) = separated(0)
            userages(counter) = separated(1)

            counter = counter + 1

        Loop

        FileClose(1)




        For index = 0 To counter - 1

            txtOutput.AppendText(usernames(index) & vbTab & userages(index) &
vbNewLine)

        Next

    End Sub
```

This array will store the values separated from a single CSV line. The index number should be the same as the number of items per line.

Split places each item into a new position in separated each time a comma is found.

Separated items are now placed into the correct array

Practise Tasks

1. Create a program that asks 5 users to enter their first name, surname and age. The program should store this information in a file.

   The user should have the opportunity to display the stored information by clicking a button.

2. A teacher requires a program to store test scores for pupils in a Computing department. The program should ask for the pupil name and test score.

   The test is out of 60 and the program should also calculate the percentage score for each pupil.

   Pupils do not sit tests on the same day so results should be entered individually and stored in a file.

   At any point, the teacher should be able to recall all stored results. At the start of each session, the teacher would like to clear the stored data.

3. Create a program that collects the rainfall figures (in millimetres) for each month. Details of the month name and rainfall will be entered on a monthly basis.

   The program should also determine whether the rainfall is low (below 5mm), medium (5mm to 10mm) or high (above 10mm) for each month.

   Each month, the details entered should be added to a file.

   The user should have the option of restoring the saved data at any point. When this information is displayed, a message indicating the average rainfall for the year so far should also be displayed.

Practise Questions

Question 1 (SQP Qu 20c)

A science department has 120 candidates taking courses in biology, chemistry and physics. The school wishes to identify how many candidates gained a grade ‘A’ in all three sciences and to save their names to a separate file. An extract of the data is shown below:

…

Ann Smith,A,B,B

Peter Irwin,B,C,A

Dan Wu,B,B,C

Stacey Williams,A,A,A

Callum Reid,A,F,B

Kevin Richardson,A,A,A

…

The top-level design for the program is shown below. 1. Get details from file 2. Find and count names of students with three As 3. Display number of students with three As 4. Save three As in file

Using a recognised design technique, refine step 4. (4)