

# COMPUTATIONAL METHODS

## Visual Basic Programming – Demonstration Program 10

```
' csvanal - demonstrate reading and analysis of CSV files from console
'
' include to support StreamReader and TextFileParser
Imports System.IO
Imports Microsoft.VisualBasic.FileIO

' class to hold details of each column
Class Column
    ' name of column
    Public name As String
    ' list of unique tokens with counts, uses SortedList object
    Public tokens As SortedList
    ' create a new column
    Public Sub New(ByVal nm As String)
        name = nm
        tokens = New SortedList
    End Sub
End Class

Module Module1
    ' list of columns
    Dim ctab As New List(Of Column)
    ' main program - process command line
    Sub Main(ByVal Args() As String)
        If Args.Length > 0 Then
            ' assume argument is file to process
            Dim sr As StreamReader = New StreamReader(Args(0))
            Process(sr)
            sr.Close()
        Else
            ' no arguments, so process standard input
            Dim sr As StreamReader = New
StreamReader(Console.OpenStandardInput)
            Process(sr)
            sr.Close()
        End If
        ' produce summary
        Summarise()
        ' that's all folks
        Environment.Exit(0)
    End Sub
    ' process a CSV file, updating column table
    Public Sub Process(ByRef sr As StreamReader)
        ' assign a text field parser
        Dim tp As New TextFileParser(sr)
        ' set it up to look for fields separated with commas
        tp.TextFieldType = FileIO.FieldType.Delimited
        tp.SetDelimiters(New Char() {","})
        ' process the file row by row
        Dim currentRow() As String ' array of text fields
        Dim rcount As Integer = 0 ' row count
        ' process until end of file
        While Not tp.EndOfData
            Dim ccount As Integer = 0 ' column count
```

```

' get the fields in this row
currentRow = tp.ReadFields()
' loop through all the fields
For Each currentField As String In currentRow
    If (rcount = 0) Then
        ' assume first row contains column names
        ctab.Add(New Column(currentField))
    Else
        ' if this is a new column, initialise column record
        If (ccount > ctab.Count) Then
            ctab.Add(New Column("Unknown"))
        End If
        ' add contents to sorted list
        If (ctab(ccount).tokens.Contains(currentField)) Then
            ' already there - bump value counter
            ctab(ccount).tokens.Item(currentField) += 1
        Else
            ' not there yet, add it with value of one
            ctab(ccount).tokens.Add(currentField, 1)
        End If
    End If
    ' bump the column counter
    ccount += 1
Next
' bump the row counter
rcount += 1
End While
End Sub

' summarise contents of column table
Public Sub Summarise()
    ' for each column entry
    For Each col As Column In ctab
        ' report column name
        Console.WriteLine("Column=" & col.name)
        ' report each of the list entries
        Console.WriteLine("Values=")
        For Each de As DictionaryEntry In col.tokens
            Console.WriteLine(vbTab & de.Key & vbTab & de.Value)
        Next
    Next
End Sub
End Module

```

```
Mark@xps /cygdrive/c/src/vbasic/csuanal
$ cat test.csv
subject,attempt,expt,result
S1,1,TEST,1
S1,2,COND1,4
S2,1,COND1,3
S3,1,COND2,3
S3,2,COND1,2
S3,3,COND1,1

Mark@xps /cygdrive/c/src/vbasic/csuanal
$ bin/Release/csuanal test.csv
Column=subject
Values=
      S1      2
      S2      1
      S3      3
Column=attempt
Values=
      1      3
      2      2
      3      1
Column=expt
Values=
      COND1  4
      COND2  1
      TEST   1
Column=result
Values=
      1      2
      2      1
      3      2
      4      1

Mark@xps /cygdrive/c/src/vbasic/csuanal
$ _
```