CIS 152 – Introduction to Computer Programming for Business

Week 7 – Data Files (Sequential Text)

1. Files
   1. We will be dealing with sequential text files. This means the content will be text as opposed to binary. We will access the file sequentially meaning, we will always start at the beginning of the file and go to the end.
   2. You will need to change your .gitignore file otherwise you text files won’t get saved in gothub
      1. Open .gitignore in a text editor (notepad)
      2. Go to the section that starts with   
         # Build results
      3. Put a # (git comment) in front of every line down to   
         # Visual Studio 2015/2017 cache/options directory
      4. You probably also want to make sure your log file and your configuration file extension is txt
2. File Access (in examples sw and sr are just sample variable names)
   1. Write – use a data type of StreamWriter
      1. Create
         1. The file is created new every time. All data in an existing file is deleted.
         2. Syntax: StreamWriter sw = File.CreateText(NameOfFile)
      2. Append
         1. The text is added to the end of the file. If no file exists one is created
         2. Syntax: StreamWriter sw = File.AppendText(NameOfFile)
      3. To write to a file use sw.Write(text) or sw.WriteLine(text)
   2. Read – use data type StreamReader
      1. The file can only be read.
      2. Syntax: StreamReader sr = File.OpenText(NameOfFile)
      3. In addition care must be taken to make sure file exists (see below)
   3. In all cases above:
      1. File must be closed sw.close or sr.close
      2. The declaration does not (and in fact should not) be part of the opening of the file
3. Exception handling
   1. When a problem can happen (like a file that is supposed to be there isn’t) Programers need to take steps to make sure the program handles them gracefully (doessn’t crash)
   2. To do the we use Try..Catch..Finally blocks
      1. We surround possible problematic code with a Try block.
      2. The catch block is the code that runs if something bad happens.
      3. A finally block is optionally added if you have code that after both the try and the catch.
4. FileOpenDialog
   1. This is Windows provided code that allows us to give the user a dialog to choose a file to open.
   2. This should be added to the form before it is used
   3. Toolbox->Dialogs->OpenFileDialog
   4. You only ever need one of these at a time, so you can keep the default name if you wish