File Commands

        During the course of running a program, the data that is used is found only in the working memory of the computer. Once the program is closed, all the data that is not in the actual code (i.e. the data that is inputted by a user) will be lost. In order to access data between sessions or to use data that is initially needed by the program, you must use files.

Opening Files

To use any file for reading or writing, the program must first open it for access.

        **FileOpen (*<file number>, <filename>, <OpenMode>)*

- *File Number* (i.e. 1, 2, 3, etc…) is a numeric reference to the file. This number will be used by other commands and predefined functions for files.
- *Filename* (i.e. "c:/students.txt") can either be a string in quotations ("") or a string variable or constant.
- *OpenMode:* There are 3 Basic Modes for opening a file depending on how that file is going to be used.
  - Reading/Input Mode:
  - **OpenMode.Input -** needs an already created and formatted file to open properly
  - Writing/Output Mode:
  - **OpenMode.Output** - will create a new file or <u>overwrite</u> an existing file
  - **OpenMode.Append** - will add data to the end of an existing file rather than overwriting the entire file

Reading/Inputting

Visual Basic's Reading/Input function will notice 3 basic symbols that are in every file.
- **EOF**: symbol that marks the very end of a file.
- **EOL**: symbol that marks the end of every line in a file.
- **Commas (",")**: Can be used to distinguish between different pieces of data in the file.

        To read in a piece of data into a variable, the data items in the file must be separated by:
            Line:
                Sam
                Jane
                John
            Or Comma:
                Sam, Jane, John
            Or a combination of both:
                Sam, Jane, John
                Jack, Jill, Tom

        Input (*<file number>, <variable>*)

        This command reads one item of data from a starting point to a comma or EOL. The starting point is either the beginning of the file or the last place where an

Input command stopped. The *file number* refers to an already open file designated for input.
(Note: You can only read in ONE variable per Input command). The data will be converted to be the type of the variable.

There is another function that will read the entire line as a string. This is useful, for example, if you want to get a line of text from a file that has commas in it. We will not emphasize this function in this course.

<string> = **LineInput (*<file number>*)**

The *file number* refers to an already open file designated for input.

Writing/Printing

Unlike the **Input** command you can write/print multiple variables to a file under a single **Print** command. This can be done one of two ways:

Print (*<file number>*, *<variable>*, *<variable>*, *<variable>*, …)
-or-
Print (*<file number>*, *<variable>* & " " & *<variable>* & …)

The *file number* refers to an already open file designated for output.
(**Note: In the second way the '&'** concatenates multiple variables and other strings (spaces or commas) and other string functions (i.e. vbcrlf) into one big string.)

There is also a command called PrintLine that behaves the same as Print except that it automatically puts a newline character after all the values that you give it to print to the file.

Testing for End of File

The end of file marker can also be detected by a predefined function, also called EOF. This function returns a Boolean which will be true when the EOF marker is reached in the file.

```
While Not EOF(<file number>)
        Input …
End While
```

Closing Files

When the user ends the program, all open files will be closed. But sometimes when writing/printing to files the information that has not yet been written to the hard drive will still be stored in the buffer of the computer and the file will not be complete. So get in the habit of <u>ALWAYS CLOSING EVERY FILE THAT YOU HAVE OPENED</u>. You can close any file anywhere in the program, but just make sure the program closes it before it Ends.

FileClose (*<file number>*)

(**Note: If you do not use a *file number* and simply put **FileClose()**, the program will close <u>every</u> file that is open.)