Using Data Files

- To this point, we have used the keyboard to input data into our programs and the output from the program was displayed on the monitor.
- The problem with this is that the data is gone forever once the program is finished executed.
- Often, we wish to save the information we use for a long period of time. This is done by saving the data to a data file stored on a hard drive or another long term storage device .
- Data inputted into the program will come from a data file instead of the keyboard. Data outputted from the file will be sent to a data file instead of the monitor.

Opening a File Using the *open* Function

- In order to re-direct input and/or output in a program, the first thing that must be done is to create a "path" to connect the data file to the program file.
- This path is identified by a **stream number** which is stored in string variable.
- Every file which is opened must have a separate stream number.
- Note that creating this path does not actually move any data.

- This path should be established near the beginning of the program and definitely before the file is accessed.
- This is done using the open function as in:

  streamnumber = **open (**"filename", “capability”)

  - *streamnumber* is the name of the variable which stores the stream number of the data path (sn, snIn or snOut are often used as the name of the variable).
  - *filename* is the name of the data file. Note that it must be surrounded by quotes unless it is a variable name. If the data file is not in the same folder as the program, the entire path must be entered.
  - *capability* tells the computer which way the data is moving and can be one of:
    - **r** which is used to open the file for reading (inputting)
    - **w** which is used to open the file for writing (outputting), and any file with the same name is erased - be careful!
    - **a** which is used to open the file for appending (outputting) where data written to it is added on at the end of the file.
    - Note the capability must be enclosed in quotations (“”).

The close () Function

- This function is used sever the link between the data file and the program.
- In Python, this is supposed to be done automatically when the program has finished executing but this is not done at times.
- In some other to programming languages, it is never done automatically.
- To insure all programs work, this function should always be included at the end of the program.
- It must also be used when changing the direction of data flow (ie. inputting data to and outputting data from the same file).
- The form is ***streamnumber*.close** () where *streamnumber* is the streamnumber variable

Inputting Data Into the Program From A Data File

''' A program to input all the sentences from a file called data..txt and output them on the monitor. Note that a for loop cannot be used because the programmer usually can't be sure how many pieces of data are in the file'''

```
snIn = open ("data.txt","r")          # Opens a path to the data file
                                      # No data is inputted at this time

while True :                          # Loop continues until the break command is executed

    data = str.strip (snIn.readline ())   # Inputs the name of the student from the file
                                          # The Python function called readline () inputs a single line of data #from a file. It must
                                          # be "attached" to the streamnumber variable with a  period (.)
                                          # Unfortunately, the computer often adds "invisible codes" like line returns which could
                                          # mess up the look of the data. To eliminate these extra lines, use the str.strip function

    if data == "":      # This will exit the loop using the break command if name is a null string (end of file)
        break

    print (data )                     # Output data to the monitor

snIn.close ()   # Severs the link to the data file. Note it must be outside the loop.
```

Outputting Data To A Data File From the Program

Example 1 - The programmer knows how many names to output and the data file is created

 # A program to input 3 names and output the names into a file called "data.txt"

```
snOut = open ("data.txt","w")         # Opens a path to the data file
                                      # No data is outputted at this time
                                      # Note that any current file called data.txt is deleted without warning.
for i in range (0,3):  # Note that a while loop can be used as well
    sentence = input ("Enter a sentence ")
    snOut.write (sentence + "\n" )    # Outputs the contents of sentence to the file identified by the streamnumber snOut using the
                                      # write function. The write function is used instead of print() function to output data to a file.
                                      # Note that a newline character ("\n") must be concatenated to the inputted string or all the
                                      #  data will be outputted on a single line in the file.

snOut.close () # Severs the link to the data file. Note it must be outside the loop.
```

Example 2 - The programmer does not know how many names to output and the data file is <u>appended</u> (added to)

 # A program to input any number of names and output the names into a file called "data.txt"

```
snOut = open ("data.txt", "a")      # Opens a path to the data file
                                    # No data is outputted at this point
                                    # Any other file with the same name is added to (appended)

while True:
        words = input("Enter a sentence")
        snOut .write (words + "\n" )    # Outputs the contents of sentence to the file identified by thestreamnumber snOut using
                                        #  the write function. The write function is used instead of print function to output data to a file.
                                        # Note that a newline character ("\n") must be concatenated to the inputted stringor all the data
                                        # will be outputted on a single line in the file.
        ans = input ("Do you wish to input another sentence (Y or N)") # Asks the user if they wish to continue inputting data
        if ans == "N" :     # Checks for both 'n' and 'N'
                break
        elif ans == "n" :
                break

snOut.close () # Severs the link to the data file. Note it must be outside the loop.
```