

## Chapter 9 - Lecture 1

ENGR1120 - 800 - Honors Programming for Engineers

April 14, 2020

**Data Input from .csv Files**

## Lecture 1 - Data Input from .csv Files

- Data Files
- File IO
- File Input from .csv
- Example in MATLAB

# What is a Data File?

## What is a Data File ?

- Standard way of organizing data for computer **storage**
- The data can represent many different things but it is all stored **digitally**
- Different **file types** are used for different purposes
  - 
  - 
  - list of all...

## Why use Data Files?

### Why use Data Files ?

- Organize large amounts of information
- Share large amounts of information
-

# What is File IO?

What is File IO?



# File Input in a Program

## File Input in a Program

- get data from a file during Program Execution
- data can be stored in a variable(s) to be used by your program

## Comma Separated Values

The individual values in a file are often separated or *delimited* by a comma. Other characters are also used such as the space or *newline*.

A .txt file with values delimited by commas is called a comma separated value file or **.csv** file.

Traditionally the *end of file* was marked by a special character as well but modern data files are organized by file size and do not require an end of file character.

## .mat and MATLAB

**.mat** files are compatible with MATLAB only

- `save()` - saves the workspace to a **.mat** file
- `load()` - loads a **.mat** file into workspace
- this is useful if you want to be stuck in MATLAB...
- a *universal* filetype is a good idea for data storage



# Universal Filetypes

Standard file formats are compatible with many software systems.

Software	File Type
C++	.txt
MATLAB	.mat .dat
Python	.csv
MS Excel	.xlsx
ilearn	.pdf .docx
Solidworks	.sldprt .stl

Some don't play well with others.

## MATLAB functions for file IO

There are different ways to get data from a file. We are going to **scan** the data one character at a time using these functions.

```
fopen()
```

```
fscanf()
```

```
fclose()
```

# The fopen() Function

Open the file with the **fopen()** function

```
[FID]=fopen(FILENAME , PERMISSION)
```

- Input 1: FILENAME - the name of the file to open
- Input 2: PERMISSION - direction of access 'r' or 'w'
- Output 1: FID - the file identifier

## The File Identifier

The **file identifier** (FID) gives important info

- If the file opens properly the FID will have a positive value
- The FID will have a negative value if there was an **error**
  - File is not in the proper **directory**
  - The **current folder** has not been set properly
  - Please organize you file structure!
- FID can also give information about the **End Of File**

## The fscanf() Function

**fscanf()** can access the data only if the file is open

```
[A , COUNT] = fscanf (FID , FORMAT , SIZEA) ;
```

- Input 1: FID - the **file identifier** fid  
Input 2: FORMAT - format specification of the scan  
Input 3: SIZEA - number of values to be scanned
- Output 1: A - an array containing the scanned **data**  
Output 2: COUNT - the number of elements in **A**

## The fclose() Function

Remember to close the file with **fclose()**

```
[ST]=fclose(FID)
```

- Input 1: FID - the **file identifier** fid
- Output 1: ST - status of close?
- Close the file after your program accesses the data
- THIS IS EASY TO FORGET BUT IMPORTANT!!!

## A Simple Example

```
FID = fopen('input_data.csv','r');  
  
A = fscanf(FID,'%f')  
  
fclose(FID);
```

## A More Complex Example

```
fid=fopen('lab9_degrees.csv','r');  
  
i=1;  
while ~feof(fid)  
    data(i)=fscanf(fid,'%f',' ',1);  
    i=i+1;  
end  
  
fclose(fid);
```



## References

- Your MATLAB textbook - Chapter 9 - Low Level File IO