ENGR 1120 Lecture Chapter 3 - Command Window Input and Output

- **basic command window output ;**

    - by default each line of code (expression or assignment) will print to the C.W.

    - however the *semicolon* is used to *suppress* the output

    - **example:**

    ```
    >>x=10;
    ```

    - the default output can be configured using the format command

    ```
    >>format compact
    >>format long
    >>format loose
    >>format short
    ```

    note: I will not be using the *format* command, you can look into if you wish

- **2 new useful functions**

  - the *fprintf()* **function**

    * similar to the C library function

    * complete control of the *command window output*

    * also used with *file output*, formatted **print** to **file**

    * results in a formatted *string*, character level control

    * **example:**

      The lines below goes in your code.

      ```
      my_var=97.563;
      fprintf('The value is %f ',my_var)
      ```

      The following output will appear in the command window.

      ```
      >>The value is 97.563
      ```

      Can you tell what happened?

– **more about the** *fprintf()* **function**

`'% feild width . precision f '`

∗ *Data Type* of the value

∗ *Field Width*

∗ *Precision*

– escape sequences and the *fprintf()* function

* `\a`

* `\b`

* `\n`

* `\r`

* `\t`

* `\\`

* `\'`

* `\''`

- the *input()* function

    * adds a simple user interface to your program

    * you type *command window input*

    * the input can be stored as a variable

    * **example:**

    The line below goes in your code,

    ```
    x=input('Please type A number ')
    ```

    and then the text will appear in the command window.

    ```
    >>Please type A number
    ```

    Now the user can type a value (shown in red below) followed by the *Enter* key.

    ```
    >>Please type A number 42.7
    ```

    Now you can see the value 42.7 is stored in the variable x.