EE1910 Coding Guide (Dr. Johnson)

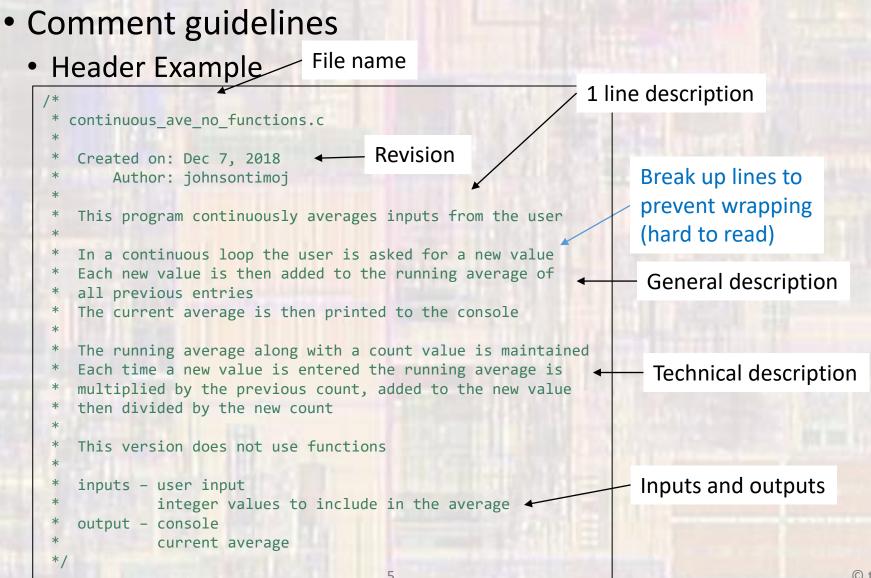
## Last Updated 12/3/20

- Windows Guidelines
  - Code Composer workspaces / projects will be placed in student directories – not inside the TI installation directory
  - Absolutely no spaces in file or directory names
    - use underscore instead: my\_project\_directory
  - File names must be descriptive
    - lab3\_part1.c or continuous\_average.c
       l3p1, try15, i\_hate\_c
    - main.c is not an acceptable program name

© tj

- General C code guidelines
  - All code must be properly indented
    - I will not help debug un-indented code
    - I will not grade un-indented code
  - Your name must be in the header section of your code
  - File structure will be: pre-processor directives function prototypes main function definitions

- General C code guidelines
  - Main will be used for control
  - Most processing will be performed in functions
    - (after we learn about functions)
  - No global variables
  - Attempt to make your code as general (re-usable) as possible
  - Use #define for constants and values used in multiple locations



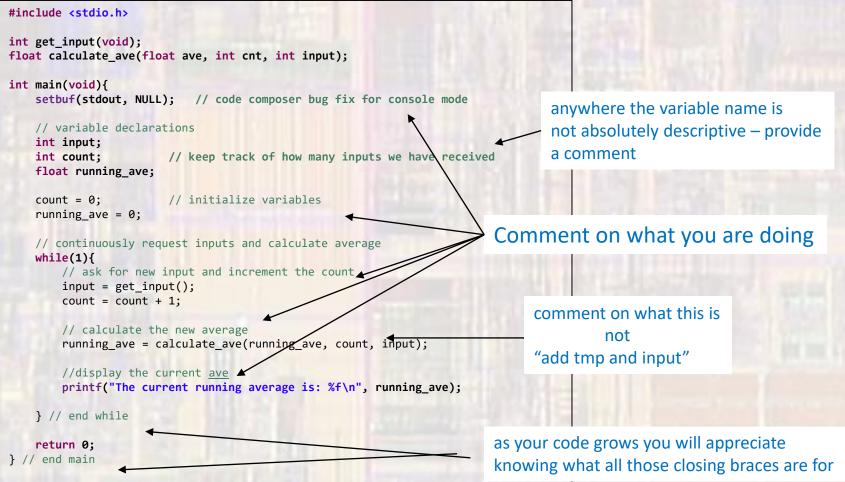
#### Comment guidelines

#### Main Example (no functions)

```
#include <stdio.h>
int main(void){
   setbuf(stdout, NULL); // code composer bug fix for console mode
   // variable declarations
   int input;
                                                                              anywhere the variable name is
                      // keep track of how many inputs we have received
   int count;
   float running_ave;
                                                                              not absolutely descriptive – provide
   float tmp;
                      // calculation temporary variable
                                                                              a comment
   count = 0;
                      // initialize variables
   running ave = 0;
   // continuously request inputs and calculate average
   while(1){
                                                                      Comment on what you are doing
       // ask for new input and increment the count
       printf("\nPlease enter an integer value: ");
       scanf("%i", &input);
       count = count + 1;
                                                                           comment on what this is
       // calculate the new average
       tmp = running ave * (count - 1);
                                              urrent su
                                                                                      not
       tmp = tmp + input;
                                            new sum
                                                                           "add tmp and input"
       running_ave = tmp / count;
       //display the current ave
       printf("The current running average is: %f\n", running_ave);
   } // end while
                                                                      as your code grows you will appreciate
   return 0;
                                                                      knowing what all those closing braces are for
} // end main
```

Main should be used for high level coordination. (primarily does function calls)

- Comment guidelines
  - Main Example (with functions)



- Comment guidelines
  - Main Example (with functions)

