File Management

Last updated 2/23/23

These slides introduce .h and .c/.cpp file management

File Management

- File inclusion
  - As a developer who has spent a lot of time developing code, I might want to allow you to use the functions in my library without giving you access to source code

  - I could give you compiled code, and the linker can include the compiled code into your final executable code
    - But you cannot see the functions and how to use them
    - The compiler cannot see the function prototypes and will generate lots of errors

  - To resolve this, I break my code into 2 parts
    - Header files – visible to you
    - Source files – ultimately these are compiled and unreadable by you

File Management

- .c files and .h files
  - .c files are used to store C code
    - Project code
    - Library code (collected functions)
  - .h files are used to store prototypes and constants
    - Function prototypes
    - Constants

File Management

- General software development process
  - Develop code using libraries from other sources along with your code
  - The owners of the libraries want you to be able to use the functions in the library but may not want you to be able to see the implementation
    - Provide xyz.h files with the prototypes (declarations) of all the functions
      - Allows you to see the format and documentation of the functions
      - Allows your code to compile without the actual xyz.c files
    - Provide a compiled version of the code (xyz.lib)
  - Your code #includes the library xyz.h file

File Management

- General software development process
  - When you ‘build’ your project
    - All of the non-excluded .c files in the project get compiled
      - This is why you can only have one file with a main function
      - The included xyz.h files allow the compiler to know what functions are coming from elsewhere
      - Compile → assemble → machine code (10110100101010)
    - The Linker then arranges all the compiled functions from all the .c files along with any pre-compiled libraries so they can be used in your program
      - Creates a single executable file

File Management

- Header Files
  - xyz.h
  - Store prototypes and constants
    - Constants
    - Structure definitions
    - Enumerated types
    - Function declarations (prototypes)

    - Wrapped in an “include guard” to prevent including the code multiple times

File Management

- Header File - Include guard
  - Prevents the same code from being included multiple times

```
#ifndef MYFILENAME_H
#define MYFILENAME_H
…
declarations
…
#endif
```

Check to see if the constant MYFILENAME_H has not been defined – #ifndef

If it is not defined,
    create the constant - #define
    execute the commands between #define
        and #endif

If it has been defined
    skip to #endif

All caps used for the constant
Based on .h file name with dot replaced by _

The constant is not initialized or set

File Management

- Header File - Inclusion
  - Header files are #included into the .c file using the code
  - Optionally they can be included into the related code .c file

Note – c system header files are
enclosed in angled brackets < >

user defined header files are
enclosed in double quotes " "