Go over Quiz #1

In small groups, describe what each line of the makefile does and answer the following questions.
- CC = g++
- exe_file = mult_div

$(exe_file): mult_div.o prog.o
  $(CC) mult_div.o prog.o -o $(exe_file)
mult_div.o: mult_div.cpp
  $(CC) -c mult_div.cpp
prog.o: prog.cpp
  $(CC) -c prog.cpp

clean:
  rm -f *.out *.o $(exe_file)

1. How could you use the variables in other places in the makefile?

2. What happens if you put the clean target at the top above the first target to make the executable?

3. What happens if you define your .o targets before your executable target?

4. What happens if you do not use tabs to indent?

Now, as a class with your TA, make these changes to see what really does happen!

As a class, discuss Assignment #1.
- What questions do you have about the assignment in general?
- What are your questions about File I/O for the assignment?

More about structs:
Let’s revisit structs and using them in functions. Given the textbook struct in Worksheet #1, how would you do the following:
- What are the three ways you could create a function called set_textbook_info() that asks the user for the textbook information and sets the members of the textbook struct accordingly?

```cpp
#include <iostream>
using namespace std;
struct textbook {
  //all represented by ints as they are whole nums
  int num_pages, year_published, edition_num, num_authors;
  char* title; //could be string, c-style in this case
  char** authors; //2D array of authors names, could be an array of C++ strings
};
```
//Define 3 different functions that will change the members in CSBook1

    set_textbook_info() {

    }

    set_textbook_info() {

    }

    set_textbook_info() {

    }

} //how would you call the three different functions?

int main() {
    struct textbook CSBook1;

    return 0;
}