# CSCD 240
# Lab 10

**Part 1**

In Java you most likely created a FileUtil class. Let's emulate that same concept in C to create a static library.

- Create an openInputFile function that takes a char pointer as the parameter, representing the filename. The function will open that file for reading and return the FILE pointer. If the file does not open, the function will call openFilePrompt(). Name your C file innoprompt.c – You will need a header file named innoprompt.h

- Create openFilePrompt function that prompts the user for the filename, the function will open that file for reading and return the FILE pointer. If the file does not exist it will continue to prompt for the filename. Name your C file inprompt.c – You will need a header file named inprompt.h

- Compile and create a static library named libinfileutil.a

- I have created a basic tester that cannot be changed. I have also provided a simple make.

- Type make and then run your code.

    - The first time pass input.txt as a command line parameter
    - Run your code a second time without any command line parameters

- You will need a header file named lab10.h that will contain the #include of your header files for your functions – In this case innoprompt.h and inprompt.h

**NOTE: ALL LIBRARIES FOR THIS LAB WILL BE 32 BIT. IF YOU SUBMIT A 64 BIT LIBRARY YOU WILL RECEIVE 0 POINTS ON THIS LAB**

## Part 2

Let's repeat question one, except this time using a dynamic library.

- Uncomment Part 2 from the cscd240_s13_lab10Tester.c
- Create a subdirectory named dynamic
- Create the openOutputPrompt function that prompts the user for the filename, the function will open that file for writing and return the FILE pointer. Name your C file openOutputPrompt.c – You will need a header file named openOutputPrompt.h
- Create the countRecords function. This function returns the count of the records in the file, it takes a FILE * and the number of lines per record as its parameters. Name your C file countRecords.c –You will need a header file named countRecords.h
- Compile and create a shared (dynamic) library named libfileutil.so.1 – NOTE: this shared library will live in the directory named dynamic
- Type make lab10_2 and then run ./lab10_2 input.txt
- Your output file will be named out.txt
- You will need a header file named lab10.h that will contain the #include of your header files for your functions – In this case innoprompt.h inprompt.h openOutputPrompt.h countRecords.h

## TO TURN IN

- A zip file containing
  - cscd240_s13_lab10Tester.c
  - lab10.h
  - makefile (this will be my makefile)
  - input.txt
  - innoprompt.h
  - inprompt.h
  - libinfileutil.a
  - the folder dynamic which contains
    - openOutputPrompt.h
    - countRecords.h
    - libfileutil.so.1

NOTE: if you give me more files than above you will receive 0 points for this lab – no negotiation / no redemption

The zip will be named your last name first letter of your first name lab 10.zip (Example steinerslab10.zip)