

# Computer Science & Engineering 150A

## Problem Solving Using Computers – Laboratory

### Lecture 13 – File I/O

Derrick Stolee

Spring 2009

- We have used `printf` and `scanf` before for input.
- These use the “files” `stdout` and `stdin`.
- Use `FILE*` type to define file pointers.

```
FILE* myFile = NULL;
```

- We can use `fopen()` to open files.
- First parameter: file name (string)
- Second parameter: file mode (string)
- Appropriate file modes:

Mode String	Mode
"r"	Read only
"w"	Write only
"a"	Append only (write to end of file)

- A use:

```
myFile = fopen("thefile.txt", "r");
```

- BE CAREFUL!
- `fopen` will return `NULL` if there was a problem opening a file.
  - File doesn't exist.
  - File is locked.
  - File is corrupted.
- Instead of crashing your program, it lets you know with `NULL`.
- DON'T FORGET TO CHECK YOUR FILE POINTER!
- After you are done, use `fclose(FILE*)`:  

```
fclose(myFile);
```

- You KNOW how to use `printf` and `scanf`.
- A slight tweak gives you `fprintf` and `fscanf`.
- Insert a parameter in the beginning for your file:

	Old	New
Out	<code>printf("%d", i);</code>	<code>fprintf(myFile, "%d", i);</code>
In	<code>scanf("%lf", &amp;d);</code>	<code>fscanf(myFile, "%lf", &amp;d);</code>

```
FILE* myFile = NULL;           /* declare pointer */
myFile = fopen("filename", "mode"); /* open */

/* check that myFile opened correctly! */

fprintf(myFile, "%d", i);      /* output */
fscanf(myFile, "%d", &i);     /* input */
fclose(myFile);               /* close */
```