file I/O (1).

- file handling involves three steps:
  1. opening the file
  2. reading from and/or writing to the file
  3. closing the file
- files in C are *sequential access*
- think of it as a cursor that sits at a position in the file
- with each read and write operation, you move that cursor's position in the file
- the last position in the file is called the "end-of-file" and is typically written as: `<EOF>`
- all the functions described on the next few slides are defined in the `<stdio.h>` header file

file I/O (2).

opening files

- `FILE *fopen( const char *filename, const char *mode );`
- `filename` is a string containing the name of the file you want to open; this file is in the current working directory or else you have to include a full path specification
- `mode` is one of the following:

mode	meaning	cursor position	create file?
`r`	read only	beginning of file	no
`r+`	read/write	beginning of file	no
`w`	write only	beginning of file	yes
`w+`	read/write	beginning of file	yes
`a`	write only	end of file	no
`a+`	read/write	end of file	no

  the last column indicates whether the file is created if it does not exist — this is only done with the `w` modes
- the function returns a value of type `FILE *`, which is a *file pointer* (we'll talk about pointers later today), or NULL if there is an error

file I/O (3).

reading from and writing to files

- these functions are just like `printf` and `scanf`, except that instead of writing to the screen and reading from the keyboard, they write to and read from a file
- for writing to a file:

```
int fprintf( FILE *fp, const char *format /*, args...*/ );
```

  this function returns the number of bytes written
  `fp` is the file pointer of the file you are writing to

- for reading from a file:

```
int fscanf( FILE *fp, const char *format /*, args...*/ );
```

  this function returns the number of bytes read
  `fp` is the file pointer of the file you are reading from

file I/O (4).

closing files

- `int close( FILE *fp );`

  `fp` is the pointer to the file you want to close (the value returned from a previous call to `fopen`)