C-style file I/O

- Rather than reading input from the keyboard (standard input), we can instead choose to read from a file
- Similarly, we can write to a file instead of writing to the screen (standard output)
- The general sequence is to get a filename, attempt to open the file, check it succeeded, perform our I/O, then close the file
- Attempts to open a file can fail for many reasons: it isn't actually a file, it doesn't exist, we don't have appropriate permissions, etc
- Filenames can even include the path to the file, e.g. `csci160/labex5/somedatafile`

Opening for input/output

- using the <cstdio> approach, out input and output file handlers will each be of type FILE*
- we attempt to open a file by specifying the name and either “r” for read mode, “w” for (over)write mode, or “a” for append mode

```
FILE *fpin;
FILE *fpout;
fpin = fopen(“somefilename”, “r”);
fpout = fopen(“anotherfilename”, “w”);
```

Checking if open succeeded

- the file handlers will be NULL (aka 0, aka false) if the open failed, so we can check simply using the ! operator

```
fpin = fopen("myfilename", "r");
if (!fpin) {
   printf("Could not open the file");
} else {
    ... it opened ok,
    ... now do stuff with it then close it...
}
```

I/O with open files

- fprintf like printf, but specify the output handle first, e.g.

```
fprintf(fpout, “x is %d\n”, x);
```

- fscanf like scanf, but specify the input handle first, e.g.

```
fscanf(fpin, “%d”, &x);
```

- fgetc same as for stdin, but specify input handle, e.g.

```
char ch = fgetc(fpin);
```

- fgets same as for stdin, but specify input handle last, e.g.

```
fgets(textArr, maxSize, fpin);
```

Checking for end of input file

- We can check for the end of the input file with feof, e.g.

```
if (!feof(fpin)) {
   // we haven't detected end of file yet
}
```

- as with the <fstream> version, the end of file isn't detected until we've attempted a read 'past' the end of the input file

Closing files

- an open file (input or output) can be closed with the fclose function, e.g.

```
fclose(fpin);
fclose(fpout);
```

Example: char by char file copy

```
#include <cstdio>

int main()
{
    char file1[7] = “in.txt”;
    char file2[8] = “out.txt”;
    FILE *fpin;
    FILE *fpout;

    fpin = fopen(file1, “r”);
    if (!fpin) {
       printf(“Could not open %s\n”, file1);
    }
    else {
       fpout = fopen(file2, “w”);
       if (!fpout) {
          printf(“Could not open %s\n”, file2);
       } else {
          do {
             char ch = fgetc(fpin);
             if (!feof(fpin)) {
                fprintf(fpout, “%c”, ch);
             }
          } while (!feof(fpin));
          fclose(fpout);
       }
       fclose(fpin);
    }
}
```