**Exercise 17: More Files**

1. This script is *really* annoying. There's no need to ask you before doing the copy, and it prints too much out to the screen. Try to make the script more friendly to use by removing features.

**from\_file, to\_file = ARGV**

**puts "Copying from #{from\_file} to #{to\_file}"**

**in\_file = open(from\_file)**

**indata = in\_file.read**

**puts "The input file is #{indata.length} bytes long"**

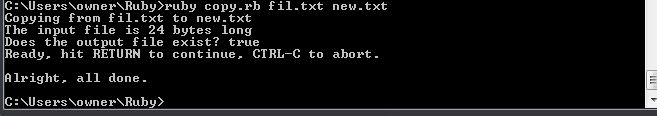
**puts "Does the output file exist? #{File.exist?(to\_file)}"**

**puts "Ready, hit RETURN to continue, CTRL-C to abort."**

**$stdin.gets**

**puts "Alright, all done."**

**in\_file.close**

****

1. See how short you can make the script. I could make this one line long.

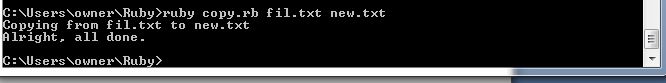
**from\_file, to\_file = ARGV**

**puts "Copying from #{from\_file} to #{to\_file}"**

**in\_file = open(from\_file);indata = in\_file.read;out\_file = open(to\_file, 'w');out\_file.write(indata)**

**puts "Alright, all done."**

**out\_file.close;in\_file.close**



1. Notice at the end of the *What You Should See* I used something called cat? It's an old command that "con\*cat\*enates" files together, but mostly it's just an easy way to print a file to the screen. Type man cat to read about it.

**Ok.**

1. Find out why you had to write out\_file.close in the code.

**Because after it has collected the object, there is no way to close the file anymore. So, closing files is highly recommended to retain manual control of your script/files behaviour, prevent unexpected behaviour caused by unpredictable output buffer flushing.**