Lab: Trapping and Killing

Our ability to delete customers was pretty well thought-out. It disallows two or more people to modify the same file by the use of a lockfile. But we missed something. What if one of those delete processes is interrupted before it can finish? Remember that the first thing we do is create the lockfile and the last thing we do is delete it. The lockfile needs to be deleted if our process is interrupted. Let's do that by trapping the interrupt signal.

1. Open the deleteCustomer function and examine it. Refresh yourself with how you created the lockfile and how you're deleting it.
2. Let's test out the interruption. Temporarily put a sleep 3 in the middle of it somewhere. Something just long enough for you to interrupt.
3. Now run it from the command line. During those 3 seconds, hit control-c to send an interrupt signal. You should get a command prompt back and if you look in /tmp, you should see your lockfile with your name and timestamp.

That's the problem we want to solve!

Trapping the interrupt signal

4. At the top of deleteCustomer, add a trap command to trap the interrupt signal. When that signal is encountered, simply print a warning to stderr for now.
5. Run and test. Make sure you see the error message. Don't forget to manually delete your lockfile again.
6. Now change your trap to do these things:
   - Print the error message as before
   - Delete the lockfile
   - Exit the script with a nonzero return code
7. Run and test. You should be able to hit control-c and interrupt your script. The lockfile should not be there.

8. Bonus! How about control-v or control-z or any other signal? Go ahead and try to trap them all.