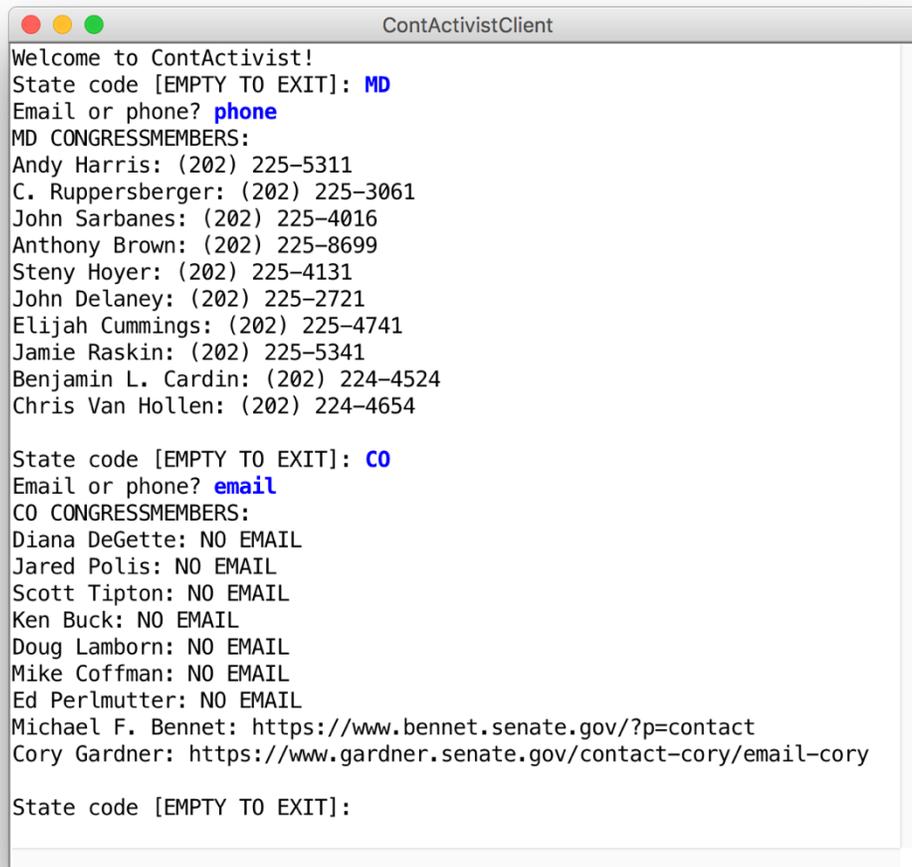


Section Handout #9: Final Exam Practice

Written by Nick Troccoli

Your task for this week is to write server and client programs for an application called **ContActivist** that allows the user to easily look up information about congress members representing different states, and how to contact them. Here is what a sample run of the client program might look like:



```
ContActivistClient
Welcome to ContActivist!
State code [EMPTY TO EXIT]: MD
Email or phone? phone
MD CONGRESSMEMBERS:
Andy Harris: (202) 225-5311
C. Ruppertsberger: (202) 225-3061
John Sarbanes: (202) 225-4016
Anthony Brown: (202) 225-8699
Steny Hoyer: (202) 225-4131
John Delaney: (202) 225-2721
Elijah Cummings: (202) 225-4741
Jamie Raskin: (202) 225-5341
Benjamin L. Cardin: (202) 224-4524
Chris Van Hollen: (202) 224-4654

State code [EMPTY TO EXIT]: CO
Email or phone? email
CO CONGRESSMEMBERS:
Diana DeGette: NO EMAIL
Jared Polis: NO EMAIL
Scott Tipton: NO EMAIL
Ken Buck: NO EMAIL
Doug Lamborn: NO EMAIL
Mike Coffman: NO EMAIL
Ed Perlmutter: NO EMAIL
Michael F. Bennet: https://www.bennet.senate.gov/?p=contact
Cory Gardner: https://www.gardner.senate.gov/contact-cory/email-cory

State code [EMPTY TO EXIT]:
```

You will need to apply your knowledge of file reading, data structures, string parsing, internet applications, and more to create this program. We have included specific details about the client and server programs below.

1. ContActivistServer

There are two types of requests that the server needs to handle:

Command	Parameters	Response
<code>getCongressPhonesForState</code>	<code>stateCode</code> (String)	Contains information about all the state's congress members. Each line contains information about a single congress member in the specified state, in the format " <code>[NAME] : [PHONE]</code> "
<code>getCongressEmailsForState</code>	<code>stateCode</code> (String)	Contains information about all the state's congress members. Each line contains information about a single congress member, in the format " <code>[NAME] : [EMAIL]</code> " If a member has no email, include the text "NO EMAIL" instead of the email.

The data comes from a file named `congress.txt`, which has the following format:

```

NAME
STATE CODE
PHONE
EMAIL (blank if no email)

NAME
STATE CODE
PHONE
EMAIL (blank if no email)
...

```

To provide a concrete example, here is an excerpt of the provided data file:

```

...
Liz Cheney
WY
(202) 225-2311

Lamar Alexander
TN
(202) 224-4944
http://www.alexander.senate.gov/public/index.cfm?p=Email

Tammy Baldwin
WI
(202) 224-5653
https://www.baldwin.senate.gov/feedback
...

```

In particular, notice how Liz Cheney does not have an email address specified, and thus the line underneath their phone number is blank. You may assume all fields other than email will always be provided.

Your server should:

- Read in the congress member information from the file `congress.txt` and store it appropriately when it begins running.
- Respond to `getCongressEmailsForState` and `getCongressPhonesForState` requests as described above.
- If a provided state code is invalid, return an error message. You may assume that if the state is not in the data file, then it is invalid.

How you implement this server, including any data structures you may choose to use, is up to you. As part of this implementation, you should also consider how designing an additional class, something like `CongressMember`, might help you in storing the required information.

2. ContActivistClient

The client should behave as in the screenshot above. Specifically, in a loop, it should:

- 1) Prompt the user for the state code they would like to look up
- 2) Ask which of either emails or phone numbers they would like to view
- 3) Print out the information for all representatives of the inputted state, including either phone numbers or emails (whichever the user specified).

The loop should terminate when the user enters the empty string (“”) as the state code.

Hint: an easy way to prompt the user for a response to an “a or b” question is to write something like the following:

```
// true if they entered "email", false if they entered "phone"  
boolean email = readBoolean("Email or phone? ", "email", "phone");
```

The `readBoolean` method takes as parameters a prompt to display to the user, followed by the “yes” answer and the “no” answer. `readBoolean` will reprompt the user until they enter one of those answers. If the user picks the “yes” answer, `readBoolean` returns `true`. If the user picks the “no” answer, `readBoolean` returns `false`.