

## Solution to Section #9

Written by Nick Troccoli

### 1. ContActivistServer

#### ContActivistServer

```
import acm.program.*;
import java.util.*;
import java.io.*;

/**
 * This server reads in data about all members of congress,
 * and can respond to "getCongressEmailsForState" and
 * "getCongressPhonesForState" requests. Both request types
 * should include a "stateCode" parameter. Both send back a string
 * containing a list of all that state's congress members, and their
 * requested information (phone or email). Note that not all members
 * are guaranteed to have an email address.
 */
public class ContActivistServer extends ConsoleProgram
    implements SimpleServerListener {

    /* The internet port to listen to requests on */
    private static final int PORT = 8000;

    /* The server object. All you need to do is start it */
    private SimpleServer server = new SimpleServer(this, PORT);

    /* The name of the congress member data file */
    private static final String DATA_FILENAME = "congress.txt";

    /* A map from state code to its list of congress members */
    private HashMap<String, ArrayList<CongressMember>> congressMap;

    public void run() {
        congressMap = readCongressFile(DATA_FILENAME);
        println("Starting server on port " + PORT);
        server.start();
    }

    /* Reads in the provided data file of congress members, and
     * returns a map from state code to a list of its congress
     * members.
     */
    private HashMap<String, ArrayList<CongressMember>> readCongressFile(
        String filename) {

        try {
            Scanner scanner = new Scanner(new File(filename));

            HashMap<String, ArrayList<CongressMember>> congress
                = new HashMap<String, ArrayList<CongressMember>>();
```

```
// Read in 5 lines at a time (for each member)
while (scanner.hasNextLine()) {
    String name = scanner.nextLine();
    String stateCode = scanner.nextLine();
    String phone = scanner.nextLine();
    String email = scanner.nextLine();
    if (email.length() == 0) {
        email = null;
    }

    // Add a new person to our map
    CongressMember member =
        new CongressMember(name, phone, email);
    if (congress.containsKey(stateCode)) {
        congress.get(stateCode).add(member);
    } else {
        ArrayList<CongressMember> personList =
            new ArrayList<CongressMember>();
        personList.add(member);
        congress.put(stateCode, personList);
    }

    // For the blank line separating each member
    scanner.nextLine();
}
scanner.close();
return congress;
} catch (IOException e) {
    println("Error reading data file: " + e);
    return null;
}
}

/* Responds to incoming requests that we receive */
@Override
public String requestMade(Request request) {
    String cmd = request.getCommand();
    println(request.toString());

    if (cmd.equals("getCongressPhonesForState")) {
        String stateCode = request.getParam("stateCode");
        if (!congressMap.containsKey(stateCode)) {
            return "Error: unknown state " + stateCode;
        }

        // Build up the response string of all member information
        String response = "";
        ArrayList<CongressMember> members = congressMap.get(stateCode);
        for (CongressMember member : members) {
            response += member.getPhoneDescription() + "\n";
        }
        return response;
    } else if (cmd.equals("getCongressEmailsForState")) {
        String stateCode = request.getParam("stateCode");
        if (!congressMap.containsKey(stateCode)) {
            return "Error: unknown state " + stateCode;
        }
    }
}
```

```
    }

    // Build up the response string of all member information
    String response = "";
    ArrayList<CongressMember> members = congressMap.get(stateCode);
    for (CongressMember member : members) {
        response += member.getEmailDescription() + "\n";
    }
    return response;
}

return "Error: Unknown command " + cmd + ".";
}
}
```

### CongressMember

```
/**
 * This class represents a single representative or senator
 * in Congress, and contains information about them including:
 * name, phone number, and optionally an email address.
 */
public class CongressMember {
    private String name;
    private String phone;
    private String email;

    public CongressMember(String name, String phone, String email) {
        this.name = name;
        this.phone = phone;
        this.email = email;
    }

    /**
     * Returns a string description of this person, which
     * includes their name and phone number.
     */
    public String getPhoneDescription() {
        return name + ": " + phone;
    }

    /**
     * Returns a string description of this person, which
     * includes their name, and email if there is one, or
     * "NO EMAIL" otherwise.
     */
    public String getEmailDescription() {
        if (email != null) {
            return name + ": " + email;
        } else {
            return name + ": NO EMAIL";
        }
    }
}
}
```

## 2. ContActivistClient

```
import acm.program.*;
import java.io.*;

/**
 * This program prompts the user for a state code, asks
 * whether they would like emails or phone numbers, and
 * communicates with the ContActivistServer to print out
 * a list of all congress members for that state and the
 * requested information about these congress members.
 */
public class ContActivistClient extends ConsoleProgram {

    /** The address of the server that should be contacted when sending
     * any Requests. */
    private static final String HOST = "http://localhost:8000/";

    public void run() {
        println("Welcome to ContActivist!");
        while (true) {
            String stateCode = readLine("State code [EMPTY TO EXIT]: ");
            if (stateCode.length() == 0) {
                break;
            }

            boolean isEmail = readBoolean("Email or phone? ", "email",
                "phone");
            sendRequest(stateCode, isEmail);
        }
    }

    /** Sends a request to get the given congress member information
     * for the given state and with email or phone information.
     * Prints out the results to the console, or an error message if
     * an error occurs.
     */
    private void sendRequest(String stateCode, boolean isEmail) {
        Request request;
        if (isEmail) {
            request = new Request("getCongressEmailsForState");
        } else {
            request = new Request("getCongressPhonesForState");
        }
        request.addParam("stateCode", stateCode);

        try {
            String response = SimpleClient.makeRequest(HOST, request);
            println(stateCode + " CONGRESSMEMBERS:");
            println(response);
        } catch (IOException ex) {
            println("An error occurred: " + ex);
        }
    }
}
```