Getting started with Program 4: Contact List application

At this point, you should be referring to the document
Last pre-break steps with Program 4: Contact List application
https://docs.google.com/document/d/1NpkrfoXq1Qmaoy-T4ENl-KLBEdR7NzpiuvBmVIdcO8w/edit#

All the relevant content of this document is included there.

In this program project, we'll be talking about the *process* of developing an application as you're building it out, so we'll be delivering the full program description later this week.

We're going to use a bottom-up approach to get started on the application.
https://www.tutorialspoint.com/difference-between-bottom-up-model-and-top-down-model

We'll need two classes for the application - Contact and ContactList - and you'll be working on the Contact class in lab this week (but you can feel free to start it before lab.)

Note: there will be additional member functions required in the final project, but adding something later will not be an issue if you know your code.

```
class Contact

    data
        char
            contactType // P(ersonal) or W(ork)

        string
            firstName lastName,streetName,city,state,email, dateOfBirth

        int
            streetNumber, postalCode

        vector of PhoneNumbers (a struct with string number and char phoneType
                                        // W(ork), H(ome), C(ell)

    constructors
        create with just a contact type, first name and last name
        create with a contact type and phone number

    basic getters and setters
        getFirstName
        getLastName
        getContactType
        getState
        getEmail
        getDateOfBirth

        setFirstName
        setLastName
        setContactType
        setEmail
        setDateOfBirth

    other "setters"
        setAddress (number, street, city, state, postalCode)


    other public member functions
        getName
            // returns ln,<sp>fn

        getAddress
            // returns number<sp>street,<sp>city,<sp>state<sp>postalcode

         string tester();   // you'll declare this but not define it;  we'll do that in the test script
```

```
class ContactList

    data
        vector of contacts

    functions

        importContactsFromFile (accepts a filename);
            loops the file, creates contacts, adds to vector

        deleteContact()

        addContact()

        string tester();   // you'll declare this but not define it;  we'll do that in the test script
```