

<https://github.com/melissaamshun/eeeeeeeeee.git>

## EE 355 Final Project- Phase I

### Team:

1. June Shao - 3502645580
2. Melissa Shun - 8500174156
3. Claire Yuan - 3545905379

### Class - CONTACT:

#### contact.h:

we made type a protected variable so that the subclasses (inherited types of the class) can access the type, since multiple subclasses have the type.

added the virtual functions to the derived class and changed the type of phone\_num to string rather than a long int because parsing strings is simpler (less bugs!) than using modulo to correct the format of the number.

#### contact.cpp:

get\_contact(): returns the contact info, either with or without type

set\_contact(): sets the type and data for the contact

print(): prints the contact by calling get\_contact()

Email() and Phone(): constructors are called and assign member variables type and email\_addr/phone\_nbr; Phone() also reformats the string phone\_number based on the size of the string (ex: 1234567890 has size -> reformat; 123-456-7890 has size 12)

### Class - PERSON:

#### person.cpp:

Person(): gets input to assign class members, allocates memory for email, phone, and birthdate

instead of deriving, Person is composed with Email, Phone, and Date class ("has a")

Person(string filename): reads from a file to get class members... uses set\_person(filename)

Person(string f\_name, string l\_name, string b\_date, string email): initializes constructor with arguments for members

~Person(): destructor frees allocated mem

get\_name(): returns a string for the full name

print(): prints the entire Person's info by calling the composite classes' print functions + get\_name()

set\_person() and set\_person(filename): takes input or reads from file, respectively, to create a new person object; distinction between email and phonenumber are handled by checking if a line in the file has an "@";

### Class - DATE:

date.h & date.cpp: templated from contact.h; added <string> standard library

added members include the numeric data (MM/DD/YYYY) and the formatted date

functions to print the date (using get\_date()) and compare between dates are added.

## Class - NETWORK:

network.h: updated with member functions, added constructors and destructor

network.cpp:

default constructor makes an empty DLL

Network(filename): uses loadDB to create a DLL from file input

~Network: deallocates memory from DLL objects

Person \*search(Person \*searchEntry): searches list by comparing a person in the list to the person referenced in the argument.

Person \*search(string fname, string lname): searches list and compares nodes with the first and last name

print\_inames(string last): traverses list and compares last names of Persons to find a match; prints if found.

loadDB(string filename): reads line-by-line through a given file and adds it to a new Person object; uses push\_front to append to list.

saveDB(string filename): prints to file by extracting data from Person and its compoa

push\_back(Person \*newEntry): used push\_front and just switched direction list propagates

remove(string fname, string lname): uses search() then removes the Person if it exists in the list by freeing the allocated mem and reattaching the prev and next pointers

## Main Menu of Class NETWORK:

1. saves the DLL to a new file using saveDB()
2. loads the network from a file; creates DDL using loadDB()
3. Allows you to add a new person by input; checks if said person already exists (no impersonation or fraud allowed here!) We check this by checking if the search function returns a null pointer.
4. removes a person by searching for their first and last name and the remove method.
5. Prints out a list of people with the specified last name
6. (not shown: used to debug; prints the entire list)

## Sources:

I used this to see how to put the variable in constructor into the variable:

<https://cplusplus.com/forum/general/13043/>

I used this to help understand operator definition:

<https://jalevine.bitbucket.io/notes/c++/2018/02/16/assignment-operators.html#:~:text=lhs%20refers%20to%20%E2%80%9Cleft%20hand,the%20right%20of%20the%20%3D%20operator.>

I used this to understand this line (cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n')):

[https://stackoverflow.com/questions/25020129/cin-ignorennumeric-limitsstreamsizemax-n#:~:text=numeric\\_limits%3A%3Amax\(\)%20sets%20the%20maximum%20number,number%20of%20characters%20to%20ignore.](https://stackoverflow.com/questions/25020129/cin-ignorennumeric-limitsstreamsizemax-n#:~:text=numeric_limits%3A%3Amax()%20sets%20the%20maximum%20number,number%20of%20characters%20to%20ignore.)

Similarly, I used this to understand that group of lines that I mentioned above:

<https://stackoverflow.com/questions/5131647/why-would-we-call-cin-clear-and-cin-ignore-after-reading-input>

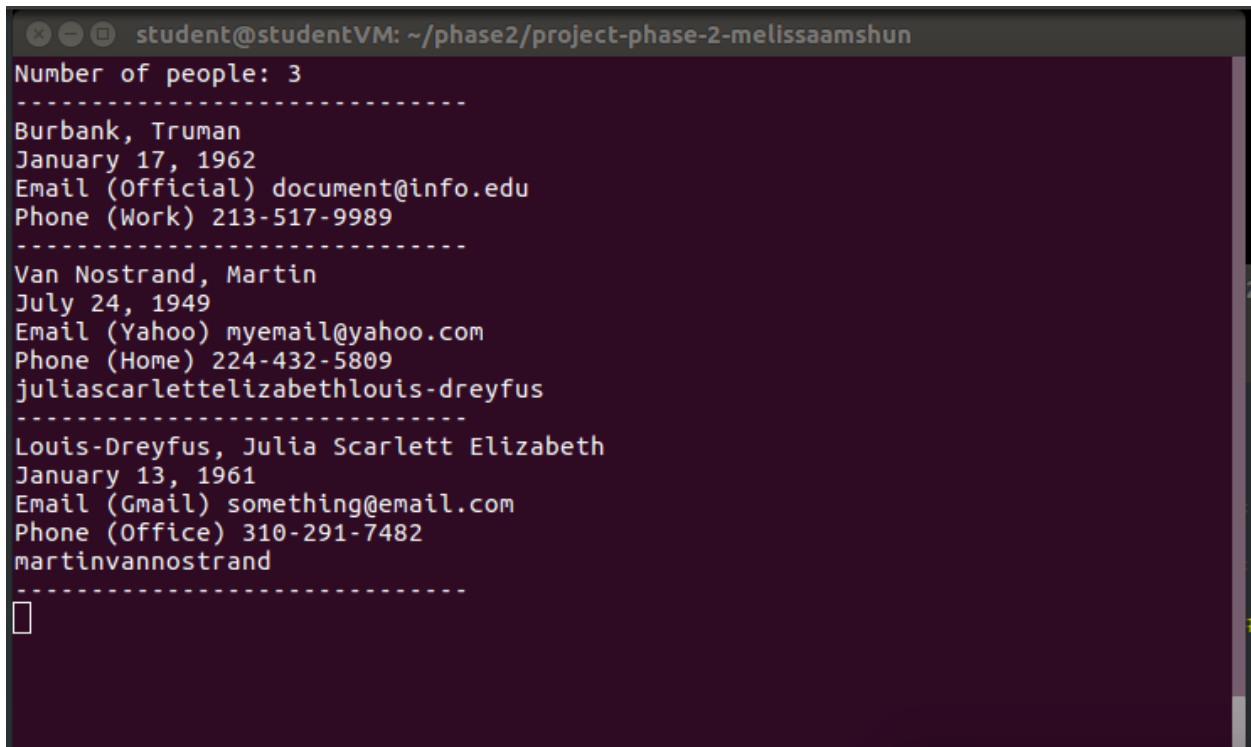
Code comments:

```
/* Note: Understand your own TA's bug!  
This code has a bug, can you understand why? no  
long int area_code = atoi(num.substr(0, 3).c_str());  
long int number = atoi(num.substr(3).c_str());  
this->phone_num = area_code*10000000 + number;  
// Guess: putting int into string  
*/
```

```
/* Note: Understand your own TA's bug!  
This code has a bug, can you understand it!  
int x1 = (phone_num/10000000);  
    int x2 = ((phone_num)/10000)%1000;  
    int x3 = phone_num%10000;  
// Guess: phone_num is string  
*/
```

Phase II !!!!!!!

Screenshots of the 'make friends' with just the loaded database



```
student@studentVM: ~/phase2/project-phase-2-melissaamshun  
Number of people: 3  
-----  
Burbank, Truman  
January 17, 1962  
Email (Official) document@info.edu  
Phone (Work) 213-517-9989  
-----  
Van Nostrand, Martin  
July 24, 1949  
Email (Yahoo) myemail@yahoo.com  
Phone (Home) 224-432-5809  
juliascarlettelizabethylouis-dreyfus  
-----  
Louis-Dreyfus, Julia Scarlett Elizabeth  
January 13, 1961  
Email (Gmail) something@email.com  
Phone (Office) 310-291-7482  
martinvannostrand  
-----  
□
```

Making friends with added ppl to the DB

```
student@studentVM: ~/phase2/project-phase-2-melissaamshun
Make friends:
Person 1
First Name:Claire
Last Name:Yuan
Person 2
First Name: Melissa Shun
Last Name:Shun

Yuan, Claire
January 26, 2003
Email (work) cyuan@usc.edu
Phone (home) 848-576-7382
juneshao

Shun, Melissa Shun
January 08, 2003
Email (school) mshun@usc.edu
Phone (null) 555-555-5555
juneshao
```

Print DB with added people and their friends

```
student@studentVM: ~/phase2/project-phase-2-melissaamshun
Number of people: 6
-----
Yuan, Claire
January 26, 2003
Email (work) cyuan@usc.edu
Phone (home) 848-576-7382
juneshao
melissashunshun
-----
Shao, June
February 30, 2000
Email (work) jshao@usc.edu
Phone (work) 000-999-2838
melissashunshun
claireyuan
-----
Shun, Melissa Shun
January 08, 2003
Email (school) mshun@usc.edu
Phone (null) 555-555-5555
juneshao
claireyuan
-----
Burbank, Truman
January 17, 1962
Email (Official) document@info.edu
Phone (Work) 213-517-9989
-----
Van Nostrand, Martin
July 24, 1949
Email (Yahoo) myemail@yahoo.com
Phone (Home) 224-432-5809
juliascarlettelizabethlouis-dreyfus
-----
Louis-Dreyfus, Julia Scarlett Elizabeth
January 13, 1961
Email (Gmail) something@email.com
Phone (Office) 310-291-7482
martinvannostrand
-----
█
```

The txt file for saveDB:

```
testsaveDB (~/.phase2/project-phase-2-melissaamshun) - gedit
Open Save
1 -----
2 Yuan, Claire
3 January 26, 2003
4 (work) cyuan@usc.edu
5 (home) 848-576-7382
6 juneshao
7 melissashunshun
8 -----
9 Shao, June
10 February 30, 2000
11 (work) jshao@usc.edu
12 (work) 000-999-2838
13 melissashunshun
14 claireyuan
15 -----
16 Shun, Melissa Shun
17 January 08, 2003
18 (school) mshun@usc.edu
19 (null) 555-555-5555
20 juneshao
21 claireyuan
22 -----
23 Burbank, Truman
24 January 17, 1962
25 (Official) document@info.edu
26 (Work) 213-517-9989
27 -----
28 Van Nostrand, Martin
29 July 24, 1949
30 (Yahoo) myemail@yahoo.com
31 (Home) 224-432-5809
32 juliascarlettelizabethylouis-dreyfus
33 -----
34 Louis-Dreyfus, Julia Scarlett Elizabeth
35 January 13, 1961
36 (Gmail) something@email.com
37 (Office) 310-291-7482
38 martinvannostrand
39 -----
```