EECS268
Lab3 write-up
Chris Watkins

1. The file format we are using is called a "comma-separated value" file or CSV. It's a common format for storing data in a text file. we used it in lab 01 as well. Is there any code from your lab 01 you can reuse to help handle the file reading? If not, explain. If so, what form does the code take (e.g. a block you copy and paste, a method, a class, etc.)
   - I reused the code from lab one that utilized the getline function with a user defined 'break' point, meaning I used char ',' to have the line separated and stored into each variable as each comma came up. I did , however, need to implement the .eof() function instead of using a passed in integer as a counter of the list length. Also, it is important to note the need to convert the strings from getline into ints for my use of these variables in subsequent functions. Below is an outline of my code:

```
while(!inFile.eof())
   {
      string tempName, tempPop, tempIFNlevel;
      char tempChar = ',';
      getline(inFile, tempName, tempChar);
      getline(inFile, tempPop, tempChar);
      getline(inFile, tempIFNlevel, '\n');
      if(inFile.eof())
      {
        break;
      }
      int pop;
      int IFNlevel;
      pop = stoi(tempPop);
      IFNlevel = stoi(tempIFNlevel);
```

2. Design a file reader class that specializes in reading CSV files. Make a list of methods you want it have. You **do not** have to actually implement this class, but if you start to think it would be handy, feel free to make a working implementation.
   - //FileReader.h

```
#ifndef FILEREADER_H
#define FILEREADER_H
#include<string>
#include<iostream>
#include<fstream>
using namespace std;

class FileReader
{
private:
  string fileName;
  ifstream inFile;
  char seperator; //could perhaps assume this to be a comma
LinkedList list; //if we wanted to store the data in a linked list
```

```
public:
  FileReader(string Filename, char seperator);
  ~FileReader( );

};

#include"FileReader.hpp"
#endif
```

3. Make a list of all methods that are called with the user chooses option 1 (increase infection rate of all cities). With each method name, please note the class it belongs to
    - upAllIFNlevel() -- MedicalExecutive.h
    - requiredActions() -- MedicalExecutive.h
    - getEntry(int i) -- LinkedList.h
    - removeCityFromList(City c) -- MedicalExecutive.h
    - decPop(int i) -- City.h
    - addFront(City c) -- LinkedList.h
    - getLength() -- LinkedList.h
    - City() -- City.h
    - addCityToQList(City c) -- MedicalExecutive.h