

Lu Yin
Instructor: Surya
EECS 268
Lab3

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.)

Yes, in lab01 I use `getline()` to read the whole file and separated each value by "comma". In this lab, the given city list's values are city name, population, and infection level. These values are also separated by "comma", so in the `MedicalExecutive.cpp`'s constructor I tried:

```
while(!inp.eof()) //keep reading
{
    std::getline(inp,e_n,','); //name
    std::getline(inp,e_p,','); //population
    std::getline(inp,e_l); //level
    City C = new City();
    C.setName(e_n);
    C.setPopulation(e_p);
    C.setLevel(e_l);
}
```

Also, in the menu optional #2, it is to choosing a city to increasing the infection level, and "Kansas City" is a hard one to save, because of the "space" included in the city name. So we tried to ignore the space:

```
Std::string str=' ';
Std::cout<<"City name:";
Std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
Std::getline(std::cin, str);
```

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 implement this class, but if you start to think it would be handy, feel free to make a working implementation.

```
Class fileReader()
{
    while(!inp.eof()) //keep reading
    {
        std::getline(inp,value,','); //separated by comma
        T value = new T();
        C.setValue(value);
    }
}

Class fileReader1()
{
    Std::string str=' ';
    Std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); //ignore the space
    Std::getline(std::cin, str); //save
}
```

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

In MedicalExecutive.class:

```
//constructor
MedicalExecutive(char*n);
//destructor
~MedicalExecutive();
//run method
void run();
//Infection level required actions
void levelActons();
```

in LinkedList class:

```
LinkedList();
~LinkedList();
isEmpty();
getlength();
positionOf(T value);
addBack(T value);
insert(int position, T value);
removeAt(int position);
setEntry(int position, T value);
getEntry(int position);
```

in City class:

```
inLevel();
getLevel();
setName(T name)
getPopulation();
setName(T name);
getName();
```

in Node class:

```
setItem(T value);
getItem();
setNext(Node<T>* n_next);
getNext();
```