Lab-5 Notes

In this lab, you will mimic the behavior of web-browser using linked list.

- For that you have been given one input.txt file. You have to read the commands one by one and execute the program accordingly.

Here is an example of the commands:

Explanation of input file has been given on wiki very well. I am just here explaining How navigate works with example, which I thought, is confusing.

Commands:

1. Back: Redirect to previous URL
2. Forward: Redirect to next URL
3. Navigate: Navigate the browser to given URL. When you encounter the Navigate command, ignore the URL accessible going forward.

For example:

Oldest

```
===========
http://google.com
http://reddit.com
http://facebook.com
http://myspace.com  <==current
===========
Newest
```

Your current URL is: http://myspace.com in above example. Now You encounter two Back, and print history you will get following:

Oldest

```
===========
http://google.com
http://reddit.com <==current
http://facebook.com
http://myspace.com
===========
Newest
```

```
VOID BROUSER :: BACK () {
    CURRENT --;
}
```

Now your current URL is : http://reddit.com. Now you encountered Navigate http://ku.edu command, then you are required to ignore what is coming after current URL, which means you have to ignore http://facebook.com and http://myspace.com. Your output should be:

```
    Oldest
===========
http://google.com
http://reddit.com
http://ku.edu <==current
===========
Newest
```

LENGTH = 4
CURRENT = 2

```
VOID   BROWSER :: NAVIGATE(STD::STRING URL){
        IF (CURRENT < HISTORY_LIST → LENGTH()){
          INT LENGTH = HISTORY-LIST → LENGTH();
          INT NEW_LENGTH = LENGTH - CURRENT;
```

For completing this lab, you have been given two interface fies.

1. ListInterface
2. BrowserHistoryInterface.

Again, as you did in previous labs, you have to derive class from the interface. The list of methods are given on wiki page for both interfaces, that you required to implement. All methods are pure virtual, so you have to have implement those in your derived class.

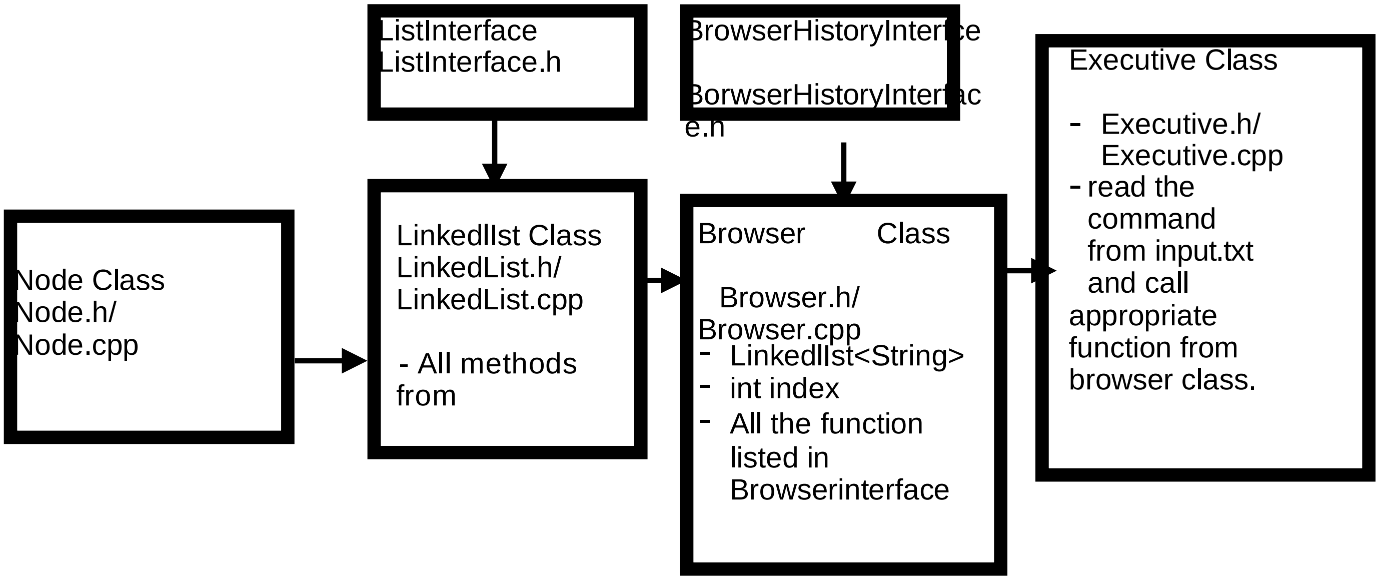
For example, you derive LinkedList class from the ListInterface and Browser class from BrowserHistoryInterface.

The main thing is how you connect this two class to fulfill the lab requirement for mimicking browser behavior.

1. LinkedList.h/LinkedList.cpp: This is very straight forward. You should continuously compile your code after writing one function, otherwise it would be very difficult to locate the error.

2. Browser.h/Browser.cpp: In this class you are required to implement all the functions listed in BrowserHistoryInterface. Now you will require two member variables, one is of type linkedlist and second is int index to keep track of current index of URL. By using these two variables, you can implement your Browser.cpp.

Here is the structure flow:



Total files:

Node.h
Node.cpp
ListInterface.h
LinkedlIst.h
Linkedlist.cpp
CurrentBroswerHistory.h
Browser.h
Browser.cpp
Executive.h
Executive.cpp
main.cpp
Makefile
input.txt