

Introduction

In this project you will design and implement linked list and linked list operations for the web browser. You will be provided with a starter template with all the test cases. You need to create linked list from scratch and need to add functions that are specified in the document. Do not change the main function, I have added test cases in it.

Web Browser for CSUF Students

The CSUF IT department is looking for potential developers who will develop the web browser which performs the same operation as browsers available in the market. They want candidates who are interested in making the web browser using a doubly linked list. The browser will implement particular functions for security and efficient use of CSUF students. Though the browser will have front end and back end. But your task will be in the backend, you have to use C++ programming to implement algorithms.

Create a class Tab which represents the node and has three values: url, name, and memory.

- url (string) - represents url of website
- Name (string) - represents name of a site
- memory (Generalised type) - represents memory consumed by site

Also, it has two pointers next and prev. Create another class Browser which contains all the necessary functions in order to implement specified functions. You need to declare two more pointers head and tail to represent Browser as a doubly linked list. You can add additional pointers to implement the following functions.

- 1) addNewTab() - This function takes three parameters- url, name and memory and creates a new tab. It will assign values to tab values and add it at the last position of a linked list. Always remember the last tab will be the current tab.

```
Browser tab list =  
| Google x|-->
```

- 2) switchToPrevTab() - This function switches the cursor from current tab to previous tab. If there is a previous tab, print the details of the tab like url, name and memory. If there is no previous tab it should print a message that says "No previous tab".

```
Switch to previous tab =  
No previous tab
```

```
Switch to previous tab =  
https://kaggle.com  
Kaggle  
310
```

- 3) switchToNextTab() - This function switches the cursor from the current tab to the next tab. If the next tab is present then print details of the tab. If there is no previous tab it should print a message that says “No next tab”.

```
Switch to Next tab =  
No next tab
```

```
Switch to next tab =  
https://www.google.com  
Google  
23.45
```

- 4) closeCurrentTab()- It deletes the current tab from the browser. When you close the current tab the new current tab will be the next tab. Print the new current tab which is the next tab with a message “Now the current tab = “ and current tab name.

Before execution:

```
Browser tab list =  
| Kaggle x|-->| Google x|-->| YouTube x|-->| GeeksForGeeks x|-->| ChatGPT x|-->| Github x|-->
```

After execution:

```
Now current tab is= Google  
Browser tab list =  
| Google x|-->| YouTube x|-->| GeeksForGeeks x|-->| ChatGPT x|-->| Github x|-->
```

- 5) bookmarkCurrent() - It bookmarks the current tab and saves the pair(name,url) in the vector of pairs if the current tab is still present in Browser. Use std::pair to store pairs in the vector. Vector can be defined as - std::vector<std::pair<data types of elements>> vector name

The repetition of bookmarks is not allowed, if a repetitive tab is trying to insert inside a vector it should give a message “The bookmark is already added!!”.

- 6) showBookmarkTab() - It will print all the bookmarks as follows:

Bookmarks:

name(url)

```
Bookmarks :  
YouTube(https://www.youtube.com)  
GeeksForGeeks(https://www.geeksforgeeks.com)  
LinkedIn(https://linkedin.com)  
Kaggle(https://kaggle.com)
```

- 7) moveCurrentToFirst() - If the current tab is still present in the browser, move it to first position.

Before function executed:

```
Browser tab list =  
| Google x|-->| YouTube x|-->| GeeksForGeeks x|-->| ChatGPT x|-->| LinkedIn x|-->| Github x|-->| Kaggle x|-->
```

After function executed:

```
Browser tab list =  
| Kaggle x|-->| Google x|-->| YouTube x|-->| GeeksForGeeks x|-->| ChatGPT x|-->| LinkedIn x|-->| Github x|-->
```

- 8) `total_memory()` - It calculates the total memory consumed by the browser in total. It will return the memory consumption.

```
Total Memory Consumption = 340.78MB
```

- 9) `deleteTab()` - It will delete the tab which is consuming the highest memory out of all the tabs. After deletion it will print the message and tab which is deleted with memory consumption.

```
Deleted element = LinkedInwith memory size = 410  
Browser tab list =  
| Kaggle x|-->| Google x|-->| YouTube x|-->| GeeksForGeeks x|-->| ChatGPT x|-->| Github x|-->
```