CMPUT 175 - Lab 4: Stacks

Submit by: Oct 5, 2020 *Demo in your lab section starting*: Oct 6, 2020

Goal: Become familiar with a new data structure: Stack.

Exercise 1:

You are tasked with creating a web browser simulator. Your web browser simulator should use **two stacks**: one to enable the **back button** functionality, and one to enable the **forward button** functionality. Your simulator will not actually display any webpages, but just the URL address of the current page that the user is on. (See sample run at the end of this exercise.) When the user wishes to enter a new webpage address, s/he signals this by entering '='. When the user wishes to go back, '<' is entered. When the user wishes to go forward, '>' is entered. The user can quit by entering 'q' when prompted.

1. Complete the worksheet at the end of this lab to show the contents of the back Stack and forward Stack after every valid entry in the sample output. **DO NOT submit your worksheet on eClass, but be prepared to show this to your TA during any help or demo sessions.**

2. Download and save stack.py from eClass. This file contains implementation #2 of the Stack covered in the lectures. **DO NOT modify or submit stack.py.**

3. Download and save a copy of lab4_browser.py from eClass. (Be sure that you save it in the same directory as stack.py.) This file contains a COMPLETED main() function which controls the flow of operation of a web browser simulation. In the following steps, you will complete the functions that this main() function calls.

4. Complete *getAction()*. This function prompts the user to enter either a '=' (to enter a new website address), '<' (back button), '>' (forward button), or 'q' to quit the browser simulation. If the user enters something other than these 4 characters, an error message is displayed before re-prompting for a valid entry. This function has no inputs. This function returns the valid character entered by the user (str).

   ***Reminder**: Be sure to write a docstring for each of your functions.

5. Complete *goToNewSite()*. This function is called when the user enters '=' during getAction(). This function prompts the user to enter a new website address, and returns that address as a string. It also updates the two stacks, as appropriate. (*Hint:* experiment with how the back and forward buttons work on a real web browser like Firefox or Chrome. After a new address is entered, can you still go forward?) Note that you do not need to explicitly return the two stacks because the Stack (as we implemented it) is a mutable object – so **bck** and **fwd** are

actually just aliases for the stacks called **back** and **forward** in your main function. The inputs for this function are the current website (str), a reference to the Stack holding the webpage addresses to go back to, and a reference to the Stack holding the webpage addresses to go forward to.

6. Complete *goBack()*. This function is called when the user enters '<' during getAction(). An error message is displayed if there are no webpages stored in the back history, and the current site is returned (str). Otherwise, the previous webpage is retrieved (and returned as a string), and the two stacks are updated as appropriate. The inputs for this function are the current website (str), a reference to the Stack holding the webpage addresses to go back to, and a reference to the Stack holding the webpage addresses to go forward to.

7. Complete *goForward()*. This function is called when the user enters '>' during getAction(). An error message is displayed if there are no webpages stored in the forward history, and the current site is returned (str). Otherwise, the next website is retrieved (and returned as a string), and the two stacks are updated as appropriate. The inputs for this function are the current website (str), a reference to the Stack holding the webpage addresses to go back to, and a reference to the Stack holding the webpage addresses to go forward to.

Sample run:

```
Currently viewing www.cs.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: 123
Invalid entry.
Enter = to enter a URL, < to go back, > to go forward, q to quit: >
Cannot go forward.

Currently viewing www.cs.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: <
Cannot go back.

Currently viewing www.cs.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: =
URL: www.google.ca

Currently viewing www.google.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing www.cs.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: >

Currently viewing www.google.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: =
URL: docs.python.org

Currently viewing docs.python.org
Enter = to enter a URL, < to go back, > to go forward, q to quit: <
```

```
Currently viewing www.google.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing www.cs.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: =
URL: www.beartracks.ualberta.ca

Currently viewing www.beartracks.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: >
Cannot go forward.

Currently viewing www.beartracks.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: <

Currently viewing www.cs.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: >

Currently viewing www.beartracks.ualberta.ca
Enter = to enter a URL, < to go back, > to go forward, q to quit: q
Browser closing...goodbye.
```

<u>Worksheet to Match Sample Run</u>:
The first 3 steps are already done to get you started…

Step 1: Web browser opened; home page displayed

current	www.cs.ualberta.ca		
back		forward	
	(empty)		(empty)

Cannot go forward, cannot go back. No change to `current`.

Step 2: Go to new site; www.google.ca displayed

current	www.google.ca		
back		forward	
	www.cs.ualberta.ca		(empty)

Step 3: Go back to previous page; www.cs.ualberta.ca displayed

current	www.cs.ualberta.ca		
back		forward	
	(empty)		www.google.ca

Complete the rest (draw extra boxes for Stack elements if needed)…
Step 4: Go forward to next page; www.google.ca displayed

current	www.google.ca		
back		forward	
	www.cs.ua.ca		(empty)

Step 5: Go to new site; docs.python.org displayed

current	docs.python.org		
back		forward	
	www.google.ca		empty

Step 6: Go back to previous page; www.google.ca displayed

current	www.google.ca		
back		forward	
	www.cs.ua.ca		docs.python.org

Step 7: Go back to previous page; www.cs.ualberta.ca displayed

current	www.cs.ualberta.ca		
back		forward	
	empty		google.ca

Step 8: Go to new site; www.beartracks.ualberta.ca displayed

current	www.beartracks.ualberta.ca		
back		forward	
			empty

Step 9: Try to go forward, but can't; www.beartracks.ualberta.ca displayed

current	www.beartracks.ualberta.ca		
back		forward	
	www.cs.ua.ca		empty

Step 10: Go back to previous page; www.cs.ualberta.ca displayed

current	www.cs.ualberta.ca		
back		forward	
	empty		beartrack

Step 11: Go forward to next page; www.beartracks.ualberta.ca displayed

current	www.beartracks.ualberta.ca		
back		forward	
	www.cs.ua.ca.		empty