COIS 2020H: Data Structures and Algorithms

Assignment 1: Web Browsing

Due: Sunday, October 17, 2021 at 23:59 EDT. A penalty of 10% per day will be assessed for each day late to a maximum of five days. Assignments will not be accepted after Friday, October 22, 2021 at 23:59 EDT.

Teamwork: Assignments are to be completed in teams of 2 or 3 members. Solo assignments or assignments with more than 3 team members will not be accepted. Select one team member to submit the assignment on Blackboard but do ensure that the names of all team members are listed.

Purpose: To gain experience implementing, using and testing classes in C#.

Introduction:

A Web browser such as Chrome or Explorer is an interface that allows us to easily navigate from one Web page to another using a few basic commands:

1) Open (**S**tart) a new tab.
2) **M**ove to a tab.
3) **C**lose a tab.
4) Move to (**V**isit) a Web page.
5) Move **B**ack to a Web page (←).
6) Move **F**orward to a Web page (→).

Requirements:

1) A new browsing session opens one tab and sets the current Web page to a default home page.
2) Each new tab thereafter also begins at the default home page.
3) Tabs are identified from left to right on the navigation bar as 0, 1, 2 …
4) The commands to open, move to, and close a tab are the characters S, M and C, respectively.
5) When a tab is closed, the current tab is set to 0. One tab must **always** remain open.
6) When a tab is opened (started), it becomes the current tab.
7) The commands to move to, back to, and forward to a Web page are represented as the characters V, B and F, respectively, and are made relative to the current tab.
8) If one cannot back from or forward from the current Web page, then the command is ignored.
9) A move (visit) to a Web page initiates a new sequence of Web pages moving forward.
10) A Web page may be visited more than once during a browsing session.
11) Commands are entered and checked in the main program.

Data Structures:

1) The **Tab** class is implemented (and tested) in **two** ways:
    a. As two Stacks
    b. As a doubly linked list (to be implemented)

   Hint: Place each implementation of Tab in its own namespace.

2) The tabs of the **Browser** class are stored in an instance of List.

3) Both Stack and List are predefined in Visual Studio.

4) The two-Stack implementation uses the following data members.

```
class Tab {
        private Stack<string> S1;    // The current Web Page is stored at the top of S1
        private Stack<string> S2;
        …
}

class Browser {
        private List<Tab> T;
        private int currentTab;
        …
}
```

5) The doubly linked list uses the following Node class. Each tab has a single reference to the current Web page.

```
class Node {
        public string  WebPage  { get; set; }
        public Node  Next       { get; set; }
        public Node  Prev       { get; set; }
        …
}

class Tab {
        private Node currWebPage;
        …
}

class Browser {
        // same as above
        …
}
```

Methods:

1) Each command triggers a method call in the Browser class which may then trigger a method call in the Tab class.

2) The methods of the Browser class include:
    a. public Browser ( ) …
    b. public void Start ( ) …
    c. public void Move (int tab) …
    d. public void Close (int tab) …

3) The methods of the Browser also include the methods below which are mirrored in the Tab class:
    a. public void Visit (string s) …
    b. public void Back ( ) …
    c. public void Forward ( ) …

4) Other methods that you will likely implement include printing out the current tab and the current Web page.

Testing:

1) Write out in a separate document your test cases.
2) Ideally, this document is prepared before coding begins. It helps to ensure that your implementation meets the requirements.
3) Remember to include testing for erroneous inputs (e.g. incorrect tab number, incorrect command).
4) Carry out your tests once the coding is finished.

Submission:

1) Submit your source code, executable code, test document and testing on Blackboard.
2) Remember to only make one submission per team.

Grading:

1)	Main Program	20%
2)	Browser Class	20%
3)	Tab Class	40%
4)	Test Document and Testing	20%

Good Luck!