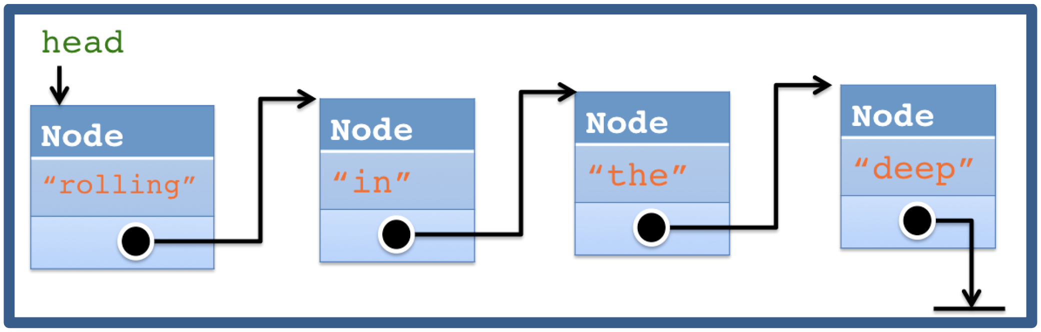
CS 200 Lab 05 Spring 2018

Introduction to Programming

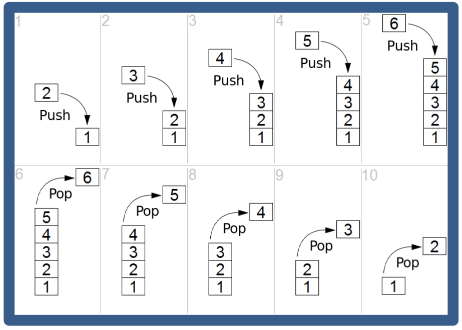
(CS200)

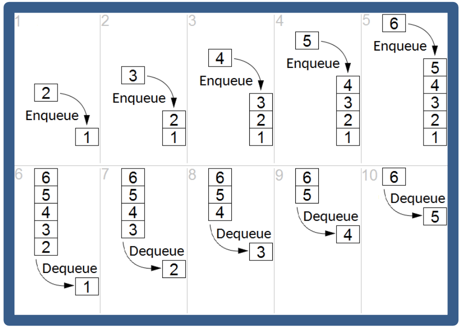
Shafay Shamail

Lists, Stacks, Queues



https://www.cse.buffalo.edu/~hartloff/CSE250/index.html?lectures/ListStackAndQueue.html





https://commons.wikimedia.org/wiki/File:Lifo_stack.png
https://commons.wikimedia.org/wiki/File:Fifo_queue.png

CS 200 Lab 05 Spring 2018

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. “Lab05”) before the time of the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.
5. Any cheating case will be reported to Disciplinary Committee without any delay.

NOTE: Define a class interface separately and its methods separately. Do not write inline code.

Marks: Name: ____________________ Roll #: ____________________

Task 1	Q1	Q2	Q3	Q4	Q5	Q6		Total
	10	10	10	10	10	10		60

Task 2	Q1	Q2	Q3	Q4				Total
	10	10	10	10				40

Let’s Begin……

Total marks Obtained

/100

CS 200 Lab 05 Spring 2018

Task 1: (60)

Implement a class called Train which has the properties of a Simple Linear Linked List. Each node of the list will be represented as a cabin of the train, with number of passengers of the train as the data in that node. Note that each cabin cannot have more than 10 passengers in it.

The following tasks need to be done:

1. Initialize two constructors, one which only creates the object and the other one which sets the number of passengers in the cabin. **10**
2.
   a. Write a class function *addCabin()* which adds a new cabin to the train. This cabin will be added regardless of whether the previous cabin has passengers less than 10. **5**

   b. Write another class function, *checkAddCabin()* which adds a cabin after checking if the last cabin in the train definitely has 10 passengers. **5**
3. Write a function to print the number of passengers in each cabin of the train. **10**
4. Now suppose that your train is going from Lahore to Karachi. There are 3 stops during the way. At each stop, 1 passenger from each alternate cabin gets off the train. Write a function *dropPassengers()* to implement this and then show the state of the train both at Lahore and when it has reached Karachi. **10**
5. Suppose during a particular journey, your train breaks down and now you need to transfer your passengers to another train. Write a class function *transferPassengers()* to implement this and show your output of both trains. The originals train should have zero passengers in each cabin. **10**
6. On the way back from Karachi, the train staff allow passengers to sit in the reverse order, such that the last cabin's passengers are now in the first cabin. Write a function *returnJourney()* to show this reversed order. **10**

STOP AND SHOW YOUR WORK TO THE TA

Task 2: (40)

Implement a class called *Stack* whose interface is given below. You can use a data structure having properties of a Simple Linear Linked List.

Add the following functions to the given *Stack* class:

1. void clear() **10**
   a. Removes all members of the stack.
2. bool duplicate() **10**
   a. Returns true if a member appears twice in the stack, otherwise false.
3. Overload the assignment (=) operator **10**
   a. Creates a copy of the given stack.
4. Overload the equality operator (==) **10**
   a. For two stacks to be equal they must have the same members in the same order.

Write a test function called *testStack()* to test the methods you have just added.

Note: Implementation of all other functions of *Stack* class may be required, however it does not carry any marks. Marks are only for the four functions mentioned above.

```
typedef int Data
class Stack
{
    private:
        Stack *sp;
        Data data;
    public:
        Stack();     //defualt constructor
        bool push(Data d); // returns true if data is pushed
        Data pop();
        bool isEmpty();
        bool isFull();
        void displayItems();
};
```

STOP AND SHOW YOUR WORK TO THE TA

Zip your tasks into one folder with format:
YourRollNo-Lab05
example "**2001001-Lab05**" and upload on LMS before the tab is closed. You will not be given extra time.