

# frankSJSU DataStructure

 Search this site

## Frank's Home Page

### CMPE126 home

### Greensheet

### Frank's Notes

[operator overloading](#)
[storing objects](#)
[pointer & deep copy](#)
[array of objects](#)
[linked list](#)
[variable size objects](#)
[create a linked node](#)
[create a linked list](#)
[linked list insertion](#)
[find middle](#)
[hybrid list](#)
[linked list quiz](#)
[recursion](#)
[stack](#)
[stack with array](#)
[math expression](#)
[queue](#)
[simulation](#)
[frankSimulation s16](#)
[priority queue & heap](#)
[search by hashing](#)

## Frank's Slides

## Frank's Code

## Programming Exam

[PE #3 F16](#)
[PE #1 guide F15](#)

## Midterm Exams

[midterm 2 F19](#)
[midterm 1 S18](#)
[midterm 2 F17](#)
[midterm 1 F17](#)
[midterm 2 S17](#)
[midterm1 S17](#)
[midterm2 F16](#)
[midterm1 F16](#)
[Labs and Homeworks >](#)

## Lab 3 Linked List

### Objectives:

1. [Get yourself familiar with the forward linked list creation and operations.](#)
2. [Manipulate pointers](#)
3. [Create linked node by is-a or has-a](#)

### Overview

1. A stock in a portfolio contains a stock symbol (e.g. AAPL), a cost (i.e. purchase price), and the number of shares.
2. Write the stock class.
3. Write a program that creates a forward linked list of at least 20 elements, where each element holds a stock. The stock symbol is a string which you can generate randomly or just do A1, A2, etc. The cost and number of shares are randomly generated between 0 and 99. Print the list. *As you may eventually find out that "reading a stock file" into the linked list is actually easier than artificially generate one. You can choose that option instead.*
4. Write the function "returnMiddleList" to find the middle element of the linked list in one pass. Print the element and the position of this element (starting at zero) in relation to the head (where the head = 0, the element pointed to by the head = 1, the element pointed to by the previous one = 2, etc).
5. Split the list in half at the middle element to create two entirely separate\* linked lists of near equal size (+/- 1) and print the two lists. In other words, write the "split in half" function.

### Discussions

1. Can you try both is-a and has-a implementation for node? What do you like or don't like?
2. If you add a length field, finding middle is fairly easy to do. What if you do not have length?
3. Do you do "add" to list when you create a new stock?
4. What if the stock portfolio is on a disk file?

### Notes:

How to convert int to string so that we can concatenate A1, A2?  
Try the simple test with stringstream:

```
stringstream s;
s << 123;
cout << s.str() << endl;
```

### Comments

[midterm S16](#)**Final Exams**[Final S17](#)[Final S16](#)[Final F15](#)[Final S15](#)**Labs and Homeworks**[Misc Lab FYI](#)[Lab 0 C++](#)[Lab 1 classes](#)[Lab 2 object array](#)**Lab 3 Linked List**[Lab 4 Doubly Linked List](#)[Lab 5 Recursion](#)[Lab 6 Stack](#)[Lab 6+ math expression](#)[Lab 7 Simulation](#)[Lab 7a Palindrome](#)[Lab 8 search](#)[Lab 9 hashing](#)[Lab 10 sort](#)

You do not have permission to add comments.

[Sign in](#) | [Recent Site Activity](#) | [Report Abuse](#) | [Print Page](#) | Powered By [Google Sites](#)