Zheng Cao <chori84@gmail.com>

CS Fundamentals Session #1 Aug 9th

Shruti Van Wicklen <shruti@adadevelopersacademy.org> Sat, Aug 11, 2018 at 12:43 PM
To: ada-cohort-9@googlegroups.com

Here's the post-session *homework* that we discussed in class on Thursday:

1. **Implement Linked List**: In the language of your choice (Ruby and/or the language you're using at work) implement the LinkedList class, Node class and all the methods we covered in class on Thursday. Bonus points for writing tests! If your singly linked list implementation is complete, do so for doubly linked list as well. Here's the Linked List github page. Ensure that your methods work when:
   - The linked list is empty
   - The linked list has only one node
   - The linked list has odd number of nodes
   - The linked list has even number of nodes
   - Head needs to be updated (delete the first node, add a node to the beginning)
2. **Revise concepts covered during classroom time**: Take some time to revise concepts we covered during classroom time. If you are leaning a new language, use the time to get well-versed with the new language: understand mutable vs. unmutable data types in the language (e.g. is string is unmutable then you cannot reverse the string in-place), understand how function calls work in the language (are parameters passed by value, by reference or by reference-value).
3. **Divide and conquer learning**: On Thursday each of you was assigned a topic to learn about based on a number assigned to you (1, 2 or 3). Take the time to learn and note down questions about this topic. Next Thursday, we'll take 15 minutes to form three groups and learn together. The week after, we'll form groups of 6 (2 students who were assigned the same topic together) and you'll share your learnings with each other. Here were the three topics:
   1. **Singleton design pattern**
   2. **Dining philosopher's problem**
   3. **Mutex** and **semaphore**
4. Explain the concepts of **Encapsulation** and **Memory leak**. We chatted about these two concepts in class on Thursday. Next Thursday, come prepared to explain these concepts as you would in an interview setting.
5. Implement using arrays and a version of insertion sort: **Median of a running stream of integers**: n integers are read one at a time from the user. Ask the user for the value of _n_. As an integer gets read, the program should compute the median of the integers read so far and display it.
   - Assume that all integers are distinct.
   - When the input size is odd, we take the middle element of sorted data to be the median.
   - When the input size is even, we pick average of middle two elements in the sorted data to be median.

Next week, we'll continue building on LinkedList interview problems and get introduced to other data structures (Stacks and Queues), Binary Search Trees.

Reach out if you have any questions or if you'd like to meet for an Office Hour on Tuesday evening (5-6pm).
Regards,
Shruti
[Quoted text hidden]
To view this discussion on the web visit https://groups.google.com/d/msgid/ada-cohort-9/CA%2BCpRvFHeNJCJT9LZQm4ddbOZz_Tm3e44nwk03h%3DyPjAPmfm_A%40mail.gmail.com.
[Quoted text hidden]