CSC 325 Adv Data Structures

<u>Program #1</u>

Java Intro

Your first assignment is a little different from assignments you have had in the past. It seems as though some students are perfectly content with grabbing bits of code online and pasting it all together to make a program, then submitting that. Whether this goes over the line into academic dishonesty (aka cheating) or not, either way it doesn't prove that you are a programmer. In fact, if all you can do is grab bits of code here and there, pretty them up, and then submit that, you have about as much right to call yourself a developer as a person who reheats frozen dinners can call themselves a chef! I know that sounds harsh, but it is the truth and this is something you need to hear now rather than wait until you are in your capstone course or in industry and still can't code from scratch. Real developers code from their head and only grab stuff online every now and then. Applications I develop myself tend to be about 85% to 95% custom written code from my head and only about 5% to 15% stuff I grabbed online. And even then the stuff I get online is mostly trivial, boilerplate things and absolutely nothing that takes real thought or imagination (if it does, I do it myself). Bottom line is, you MUST be comfortable starting with a COMPLETELY blank file and creating magic from it, with no help from online resources, past assignments, other people, any of that. If you can't do that, well, now is the time to practice!

For this assignment you will code a linked list. A simple, singly linked list. This is something you should have learned in 220 so this shouldn't be too difficult. The catch is: you will code it all on pencil and paper, in class, with no help whatsoever from ANYTHING. The linked list will allow inserting numbers, searching for numbers, and possibly more operations that I will reveal on the day of. You will only have about 30 minutes of class time to write this and I will provide the paper (bring your own pencils, or pen if you are feeling lucky). I may actually change things up a bit and tell you to make the linked list contain strings instead of ints. Or have the linked list's insertion method take extra parameters, or perhaps have the search method return the position of where it found the value. Or any number of other alterations or extra functions. The reason I do this is because I don't want you memorizing how to write a linked list. If you are going through your computer science journey here at Tech and trying to memorize coding examples, memorize exact algorithm implementations, or anything like that, you are doing it wrong. You can't memorize code. It always changes and always must be adapted. You have to <u>understand</u> the logic, <u>understand</u> the language, <u>understand</u> the algorithms, and then <u>create</u> the implementation. This may mean that it comes out slightly different each time, which is perfectly ok! Do not memorize code. It won't help you for this assignment.

To study for this, I suggest brushing up on Java by using online resources to learn and brushing up on linked lists. And then opening up a very basic text editor (a more advanced one that gives you hints will do you NO good for when you are writing this on pencil and paper in class). I suggest Notepad++ but it really doesn't matter as long as it provides very little to no help. Then, try to create a linked list. Just give it the insertion operation and maybe an operation to print all values from head to tail, so you can test it. Create the main method, instantiate the linked list, add some values, then call the print all values method you created. If it doesn't work, figure out why. If you can't, then and only then should you refer to online resources, past assignments, or any help you can find (you can even ask a friend). Now, start all over with a completely blank file and do it again, with no help. Each time you should be able to get a little further. And each time you fail, you will learn from this and will get better. Fail. Get better. Fail. Get better. And so on. Until you can do the basic operations with no help and have it work the first time you compile and run it. Failure is not your enemy. It is your teacher!

At this point, once you can code the very basics and have it work, now you should add some more interesting operations. Perhaps a prepend function that adds nodes to the front. Perhaps a function that adds nodes to the middle of the list. Add a search function. Add a delete function. Add any function you can think of. Again, try to add these without any help. If you mess up, figure out why, then start all over.

Will this take a while? Absolutely! But that is what it takes to be able to call yourself a developer. If you aren't willing to put in the time, what you are here for?

For submission:

I will give you paper on the day of. Bring your own pencil or pen (I highly recommend pencil). When time is up, you will give me your paper (with your name on it!).

Grading:

Either it works or it doesn't. College should prepare you for the real world and clients don't care how hard you tried, they only care if the application works. Fail all you want while you are practicing at home, but on the day of, failure will not be accepted.

To be clear just in case there is any confusion, I'm not looking for you to write exactly what you learned in 220. I actually don't care how you implement it. As long as it is a linked list (i.e. consists of nodes containing data and a link) and it carries out the proper operations as a linked list would (e.g. each node object links to the next). Whether your Node class has public fields or private fields with getters and setters is up to you. If I ask you to create a method that returns the size of the linked list, you could either calculate that information or keep track of a variable that is updated every time you insert or delete. Specific implementation details are up to you as long as the basics of a linked list are followed (for example, you can't use an array and call it a linked list). Remember, I don't expect you to memorize while you are learning and if you write it slightly different each time, that's ok. It just needs to work.

I want you to take this assignment very seriously. If you pull this off, this class will go a lot smoother for you.

Second chance:

I realize that some people may get nervous and make stupid mistakes. For that reason, I will offer a single second chance to anyone who doesn't nail this. If you get it wrong in class, you will need to make an appointment with me to try again in my office on another day. Note that the operations you will write will be different and, just like the first time, you won't know beforehand what they are. If you get it right this second time, you will receive a 50 out of 100 (hey, it's better than a zero). If you fail this twice, you will get a zero with no chance to try again.

Basic operations:

For the basic linked list operations, you will be asked to implement the following:

- Node class with data and link to next node
    - The data type will be given on the day of the assignment
        - I strongly encourage you to test your code with different data types such as int, float, String, etc. You might find that something that works with int might not work the same with String (the "==" operator for example).
- Linked list class with the following
    - Head pointer
    - Other fields as desired
    - Insertion method
        - To make things easy, this method will just append to the end of the linked list and will receive as a parameter the data to be appended
    - Search method
        - This method will receive data as a parameter and will return the first location of the node containing this data or -1 if not found. Keep in mind that the head node will be considered position 0, with the node after it being position 1, and so on.
    - Print method
        - This method will not receive any parameters and will just print the contents of the linked list data from head to tail. I may ask you to put a space or comma in between the node data.
- Main method
    - This can either be in your linked list class or in a separate public class. The main method should instantiate a linked list object, fill it with some data, search for a few values (some in the linked list and some not in the linked list) and then finish by calling the linked list's print method

I should be able to take exactly what you write on paper, type it into a text editor and compile and run it without any issues.

Extra operation:

I will ask you to write one additional method in your linked list class. This is not something you will know the details of beforehand. Here are some suggestions (the actual method may NOT be in this list, these are just suggestions):

- Prepend method that adds data to the front of the list
- Delete method that deletes the node or nodes with the given data
- Insert at a given location method
- Method that finds a node or nodes whose data matches some criteria
- Random insert method where a random location is generated and the given data is inserted into that random location
- Print every other node (should work for lists with both even and odd number of nodes)
- Size method that returns the number of nodes in the linked list
- SecondToLast method that returns the data in the node that's right before the tail

For all of these methods you should handle all edge cases, such as where the head is null or a particular node isn’t found.

Rubric:

#	ITEM	POINTS
1	Basic operations work	25
2	Extra operation works	25
	TOTAL	**50**