COMPX201 & COMPX241 Assignment One
Linked List

Due: Friday 8th April 11:59pm.

Part One:

Define Java classes to implement a dynamic linked list of characters (i.e. Java char) following the specification given below.

1. **The Linked List:** define a class called `ABCList` in a file called `ABCList.java`. This class is to implement a dynamic linked list of character values that support the following public methods:
    - `add(char c)` - adds to the head of the list a new node with the value of argument c.
    - `has(char c)` - returns a boolean value that is true if the linked list contains a node whose value is c; false otherwise.
    - `remove(char c)` - finds the first node whose value is c and removes that node without affecting the order of the remaining nodes in the list. If the list has no node with that value, the method should leave the list unchanged.
    - `length()` - returns (as an int) a count of the number of values in the linked list.
    - `isEmpty()` - returns boolean true if `length()` would return 0; false otherwise.
    - `dump()` - for each node in the list, print to the screen (i.e. `System.out`) its data value on a line by itself, such that all values in the list end up being printed in the order they appear in the list.
2. **The Node:** define a class called `ABCNode` for the nodes in your `ABCList`. It can either be an external class in a separate file called `ABCNode.java` or an inner class of `ABCList`. It should have the following:
    - A member variable to hold the char value.
    - A member variable to hold a link to another ABCNode.
    - A constructor that takes a char argument and copies that value into the Node's private member variable.

Part Two:

Make a copy of your `ABCList` class called `OrderedABCList` in a file called `OrderedABCList.java`. Add a new method to this class called `insert(char c)`, such that new nodes can be added in the correct place to keep your list sorted. That is, calling `dump()` on an object of this class should print all the values in its linked list in alphabetical order. Write a program to test your class.

You should now have two ABCList files (`ABCList.java` and `OrderedABCList.java)` which contain a significant amount of duplicate code. What would be the best way to remove this duplicated code? Think about object-oriented techniques and modify your solution accordingly.

A program will be posted on Moodle about a week before the assignment is due. Ensure that your classes work with this program as this will be used during marking.

Assessment:

Completing Part One can earn up to a B grade, but to be eligible for an A+ you must also implement Part Two. Your solution will be marked on the basis of how well it satisfies the specification, how well-formatted and easy to read your code is, and whether each class and public method has at least some comment explaining what it does, what it's for, and what any of its arguments are (i.e. documentation).

Your code should compile and run as a console program from the Linux command-line (i.e. no GUI). Students are encouraged to test their code in the lab prior to submitting their solutions.

Submission:

Create an empty directory (i.e. folder) using your student ID number as the directory name. Place copies of your source code (.java files) in this directory. If you wish to communicate with the marker any additional information then you may include a plain text README file, but nothing else (e.g. no compiled code (.class files)). Compress and upload this directory through the Moodle submission page for this assignment.