# OBrien, Patrick, Assignment\_1, 2019-08-06\_23-10-34:

I’d recommend that you use two separate lists – one for the history and one for the future. The front of each list is the most recent item (or the earliest item in the future) and that way you can move items from each list to the other one by interacting with only the first item in each list.

|  |  |  |
| --- | --- | --- |
| **Category** | **Mistake** | Penalty |
|  | Implementing the History class |  |
|  | PrintAll does not work  You print out everything, instead of printing out the history, then the future | -20 |
|  | PrintAll doesn’t quite work the way it does in the assignment.  You start printing from the very first item (the oldest item)– the assignment example prints starting from the most recent item and working back to the oldest item. | -6 |
|  | You don’t need a separate case for adding to the front of the list when it’s empty, and then adding to the list when it’s not empty. You can add to either list in about 4-5 lines of code | -6 |
|  | MoveBackwards does not work  It looks like you’re trying to reverse the list (?) | -20 |
|  | MoveBackwards: This will crash if the list is empty | -6 |
|  | MoveBackwards: Remove the item from the front of this list, and then add it to the ‘future’ list. You’re not adding it to the ‘future’ list | -6 |
|  | MoveBackwards: You remove the current node, take the string out, and then create a brand-new node to store the string that you add to the future list.  Change this so that you just add the same node to the future list, instead of creating a new one  Fix similar problems elsewhere in the code. | -6 |
|  | MoveForwards/Backwards: If you’re already at the end of the future/history list and you call these then you’ll add a ‘null’ string to the other list | -6 |
|  | MoveForwards and/or does not work – it’ll always reset the history to being the second item in the list. | -20 |
|  | VisitPage: you must also remove all the entries in the ‘future’ list | -6 |
|  | VisitPage: you must also remove all the entries in the ‘future’ list  First put in a comment stating (using the Big Oh notation) the running time of your current solution. Then go back and change the code that it takes constant time ( O(1) ) to clear the future list | -6 |
|  | Looks good, overall |  |
|  | Style / Presentation |  |
|  | Looks good |  |

**Grade (out of 130):**

**To calculate your grade:** add up all the (-1)'s and (-6)'s and (-X)'s, to get some negative number, then take that from the total to get your grade. For example: If the total points available for the assignment was 100, and you had the following penalties: -3 + -6 + -6 🡺 -15, so the grade would be 100 -15 = 85.

Why do you have to do this? Because this is only version 1, and so you won't really get your 'real grade' until you hand in the revision. Sometimes the grade on this first version appears really low (especially if you left out a whole section), and so I want to give people feedback, but try to avoid spooking people. Keep in mind that if you don't hand in a revision, this will be your final grade.

**Note**: Please note that if any of the above errors are duplicated within your code, you need to fix ALL INSTANCES of the error, even if it's not specifically listed above, in order to get the points on the revision.

**Note**: While the above list of errors is intended to guide your improvement of this homework, you should realize that a given error may occur in more places than have been specifically cited here. It is your responsibility to find all occurrences of a given error, and fix them all.

**Note**: Items that are 'greyed out' and *italicized* are there for informational purposes, to preemptively give you feedback for your revision, but these don't actually represent points that you've lost yet. When doing your revision make sure to pay attention to these items so you don’t lose these points in your revision. Example of a 'greyed out' item:

*You didn't do X. (-6)*