Professional Integrity Report (CPS101)

Student Name: Alex McColm **Assignment #: 3**

It took me 8 hours to complete the assignment.

These parts of the program work well:

The doubly linked list is fully generic now. It can work with any type that implements the Comparable interface, and it is tested with 2 different types in the app class. The OrderedListApp class sets up a couple of these OrderedDLinkedList objects and adds and removes data from them. The add() method inserts elements into their proper position to maintain a sorted list, and it does so efficiently. For the bonus mark, the EmployeeApp has been made to work with an OrderedDLinkedList. It is almost entirely functional: Employees can be added, searched, and the employee list can be printed.

These parts of the program don't work well (please identify the specific problem):

Removing employees is not functional in the EmployeeApp. I have uncertainties regarding algorithmic analysis (counting operations properly). I trust my results of Big O efficiency but I think I am making errors in my process to get there. The file path of EmployeeApp *may* need adjusting in order to work on another system.

I learnt the following in doing the assignment:

More of the details of using generics, editing a class and its methods to be generic after the fact. I pushed my skills in computational thinking, figuring out how a computer should do something that's intuitive to a human. I used the VSCode debugger lots, especially when retrofitting the EmployeeList to use the OrderedDLinkedList.

The difficulties I encountered were:

It was difficult to figure out the logic for the insertBefore() method and the final add() method. Initially, I used a placeholder which simply appended the element and executed an insertion sort. I figured that this method was untenable because it is inefficient, and I thought that I could probably add the element in its proper position in $O(N)$ time complexity. I spent plenty of time investigating to find out what was causing different exceptions. I had a really curious error at one point: My test was working when the Random object had a seed of 3, but not with any other seed!

Here are some other comments or suggestions: