

Philip Sloan

CS 240

Project # 1

July 8, 2014

Project one involved the implementation of the Abstract Data Type called singly linked lists. A singly linked list is a list of elements that point to another element successively in a list format. The list is constructed by creating 'Nodes' that will contain the data they are storing, along with a reference to the next item in the list. In this way, the list can have its size dynamically increased or decreased as needed. One disadvantage of a linked list is that in order to access an item in the list, the program must transverse each element of the list to reach the desired item. One advantage of the linked list is that elements can be added into any point of the list.

For this project, a singly linked list was used to create a class called Set, which would hold a set of integers, so that simple set mathematics could be carried out between sets. The Set class implemented a linked list with a dummy node at the head. Methods to add and remove elements, compare sets contents to other sets, create intersection and union sets, were created to accomplish the specifications of the project. The specifications required that the class adhere to the rules of set mathematics. For example, if set A was {1, 2, 3} and set B was {1,2,3,4} then the AsubsetOf(B) method would return true.

In order to test the class, a test program containing 5 different cases was created. Each case would be different to ensure a robust implementation of sets. Case 1 involved 2 sets that were identical but had their elements in different orders. The testing program ran the various methods of the class and printed the results. The second case was two sets that were different sizes, but one of the sets was a subset of the other. The third case was two sets of different sizes that had some elements in common, but were

not subsets of the other. The fourth case involved two sets that had neither size nor elements in common. Finally, the fifth case had one empty set, and one non-empty set. All of these cases attempted to test the various real world examples that might present themselves in set mathematics. By testing these cases, any errors could be quickly seen and corrected.

There were several lessons that I learned while programming for this project. I learned the importance of organized and thorough testing. During the testing, I found that some of my methods were causing the subset method to return true, even when it was not. Additionally, I found that I had improperly kept track of the size of the set, leading to errors when I was testing the empty set. Another lesson learned was that writing out all of the logic on paper helps to make the writing of the code significantly easier.