**Assignment 2 Grade: 101/100 (20.2/20 of the final grade)**

**Excellent job! I cannot put more than 20 points in Moodle gradebook but will keep track of the extra points in my local one and add when I calculate final course grade.**

|  |  |
| --- | --- |
| **Points** | **Criteria** |
| 10/10 | Programs use object oriented program approach (e.g. class for Queue, Stack, etc.). Programs have the appropriate naming convention, author’s name, and brief description of the implementation in each file |
| 14/15 | **DLinkedListADT:**  Correctly implements DLinkedListADT and its operations using pseudo code covered in class  Correctly implements the test program TestDLinkedList to demonstrate correctness of the code  Video correctly explains ADT and test implementation, and shows and explains running program and output   * deleteLast(): exceptions should have message * Good job with code and explanation |
| 25/25 | **DeleteFirstInstance:**  Correctly adds method to DLinkedListADT called deleteFirstInstance(data-type num)  Correctly implements algorithm to delete the first node that matches num  Correctly loads data and calls the algorithm in test program’s main method  Video correctly explains algorithm implementation, shows and explains running program and output  Video correctly explains the time and space complexity of the deleteFirstInstance algorithm   * Good job with analysis of time/space complexity * Good job with code and explanation |
| 29/30 | Correctly implements raw array **QueueADT** in single file using pseudo code covered in class  Correctly implements linked list implementation of **StackADT** in single file using **DLinkedListADT** and pseudo code covered in class  Video correctly explains implementation and operations of each ADT and shows the test programs running to demonstrate operations   * QueueADT:   + enQueue/deQueue: not good exception as nothing to do with array. You need to either create own or find existing that is meaningful * StackADT:   + StackADT(int max): not proper exception. It should be something like illegal input or argument   + push/pop: again not good exception as nothing to do with array * Good job with code and explanation |
| 23/20 | **QueueReverseMiddle:**  Correctly only uses QueueADT and StackADT and its operations  Correctly implements algorithm as method queueReversemiddle to reverse low to high element inclusively in passed in Queue instance  Correctly writes output as above to include updated queue contents in main method only using valid Queue/Stack operations  Video correctly explains the algorithm, implementation, shows and explains running program and output  Video correctly explains the time and space complexity of the queueReverseMiddle algorithm   * Time complexity: You have the correct understanding but math is off. The worst case is when you need to reverse all elements which means all putting all on stack so that’s O(n) and enqueue operation has a loop to move the elements so that means we basically have a nested loops of max size n and so we have O(n^2 + n) and so the growth rate is O(n^2). And worst space is again all elements reversed so stack needs to be size of queue O(n) * Should also throw exception when low is greater than high * Good job with code and explanation |