1. This is a doubly linked list program implemented with extra methods for programming assignment 1.
2. Author: Connor Symons
3. This project has 2 files zipped together, 1 being this README file and the other being DoubleLinkedList.cpp, which contains the Data, Node and DoubleLinkedList classes as well as an executable main program.
4. I used CLion to develop and run this file which I got for free through the university with the JetBrains Product Pack for Students.
5. Upon executing the main program, it will create an empty list and prompt you for an integer input. Inputting 1-13 as shown below will perform the associated function.

Input	Function	Description/Additional Input
1	Delete List	Deletes the list and exits the program.
2	Insert At Head	Inserts a new node at the head of the list. Will prompt for input for an integer value and a name for the new node.
3	Insert At Tail	Inserts a new node at the tail of the list. Will prompt for input for an integer value and a name for the new node.
4	Insert At Index	Inserts a new node at a specified index in the list. Will prompt for input for an integer value, a name, and the index for the new node.
5	Delete At Head	Deletes the node at the head of the list. Prints deleted node afterwards.
6	Delete At Tail	Deletes the node at the tail of the list. Prints deleted node afterwards.
7	Delete At Index	Deletes the node at a specified index in the list. Will prompt for the desired index. Prints deleted node afterwards.
8	Reverse List	Reverses the direction of the nodes in the list, makes the head of the list the tail and vice versa. Prints list afterwards.
9	Sort List	Sorts the nodes in the list in order of ascending integer value.Prints list afterwards.
10	Count Multiples	Prompts for a value and name. Returns a count of how many nodes have matching values and names to the user input.
11	Delete Multiples	Deletes all nodes in the list with matching data. Prints list afterwards.
12	Split List Head Tail	Splits the current list into two separate lists, list A and list B. List A contains the nodes from the head to the middle

		of the original list, while list B contains the nodes from the tail to the middle of the list. Odd list lengths give the middle node to list A. Prints both lists and exits the program afterwards.
13	Exit	Exit the program

6. Below is the Big O worst case run time for each of the functions explained above.

Function	Big O + explanation
Delete List	=O(n) one loop depending on length of list.
Insert At Head	=O(1) only constant time operations.
Insert At Tail	=O(1) only constant time operations.
Insert At Index	=O(n) one loop dependent on index in list (worst case is index before tail).
Delete At Head	=O(1) only constant time operations.
Delete At Tail	=O(1) only constant time operations.
Delete At Index	=O(n) one loop dependent on index in list (worst case is index before tail).
Reverse List	=O(n) one loop depending on the length of the list.
Sort List	=O(n^3) a set of nested loops is dependent on the length of the list threefold..
Count Multiples	=O(n^2) one nested loop is doubly dependent on the length of the list.
Delete Multiples	=O(n^3) a set of nested loops is dependent on the length of the list threefold.
Split List Head Tail	=O(n) multiple separate loops depending on the length of the list.
Exit	=O(1) only constant time operations.