**Data Structures Lab 8 Report**

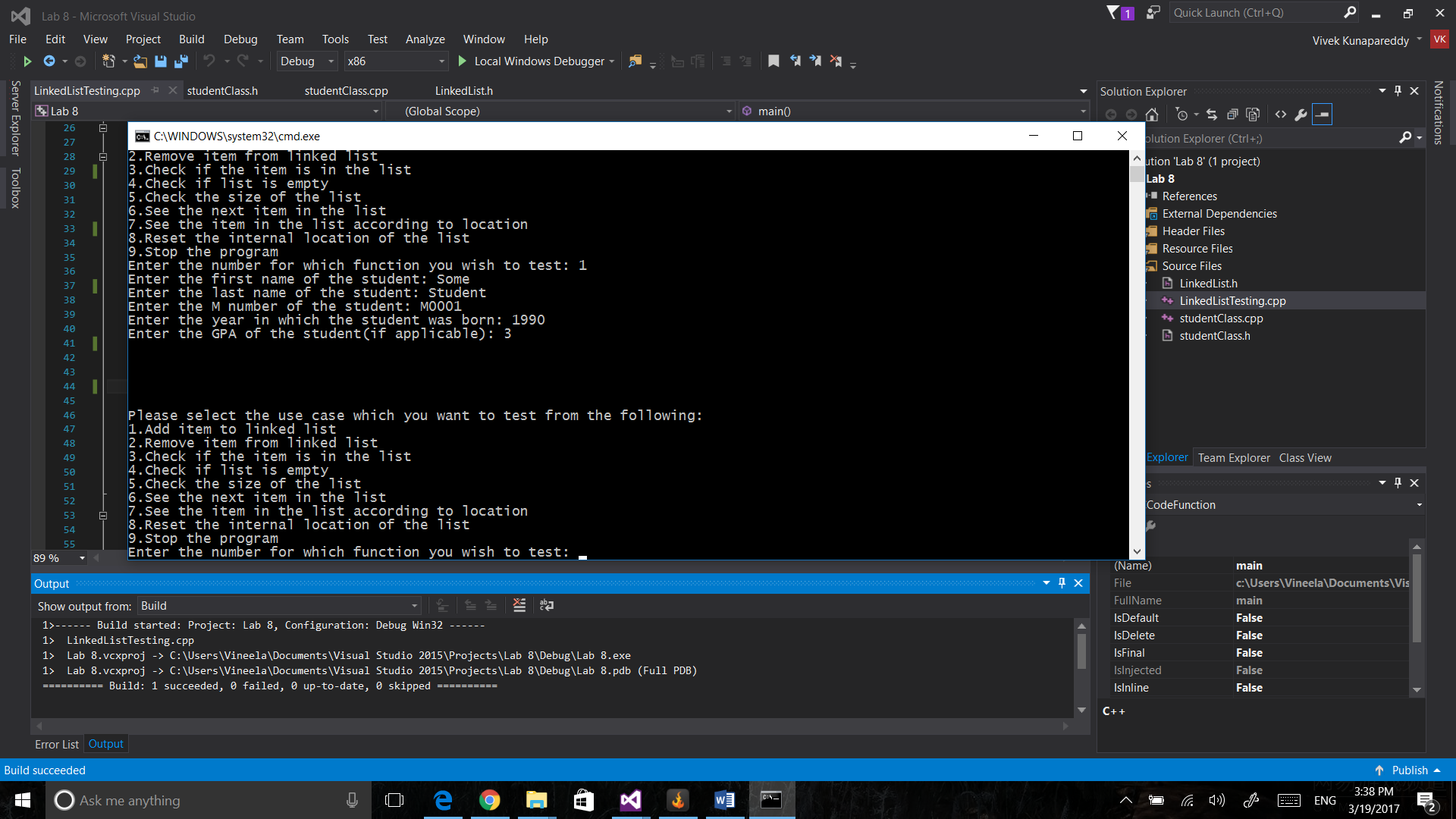
Group Members: Vivek Kunapareddy, Yuan Cheng

**Discussion of objectives/concepts:**

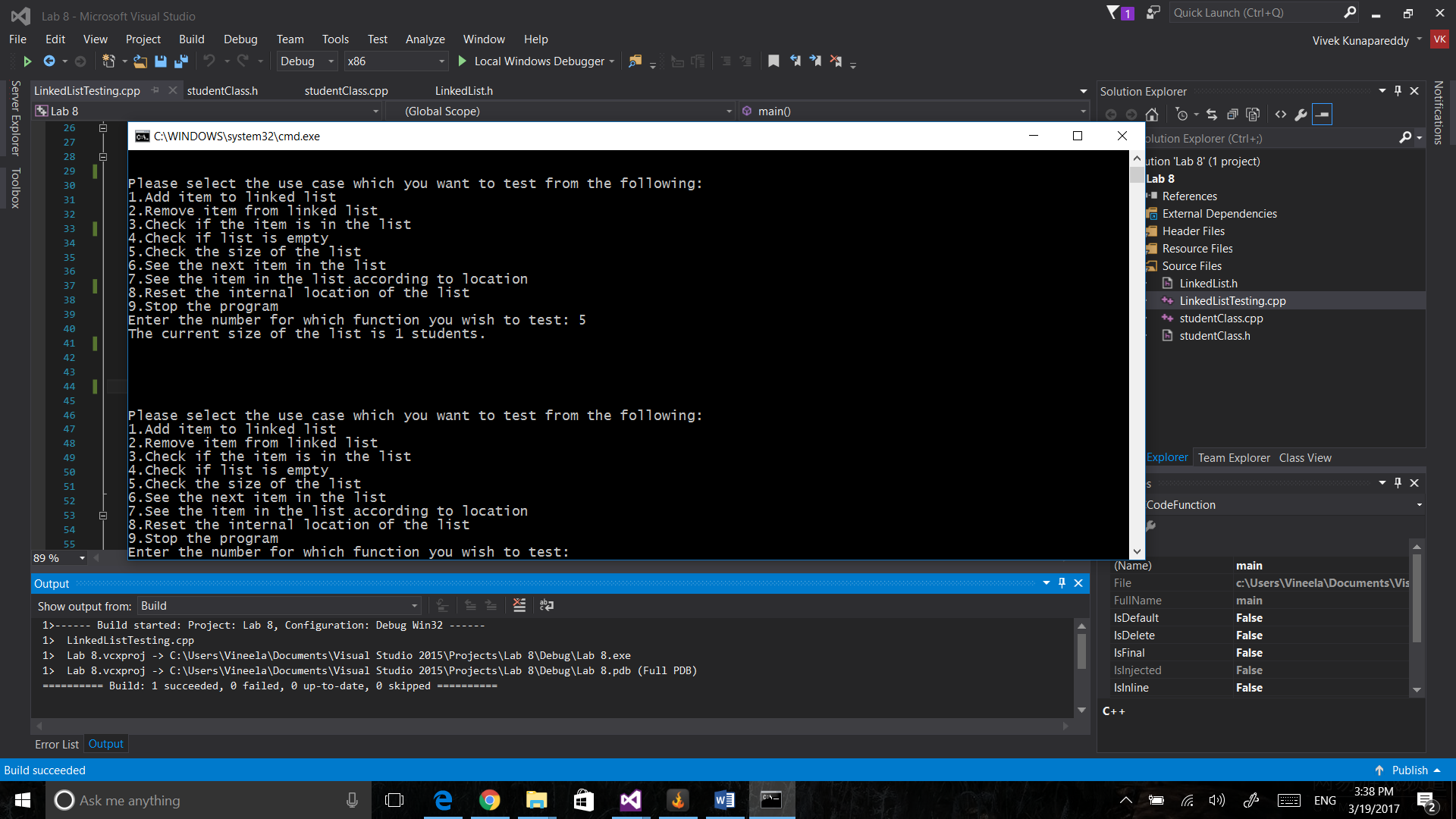
This lab assignment involved creation of an ordered linked list. This concept will be extremely useful as we transition into the software development workforce as linked lists are widely used for data storage and in multiple algorithms. Thus implementing a linked list allows us to have a stronger understanding of what goes on behind the scenes.

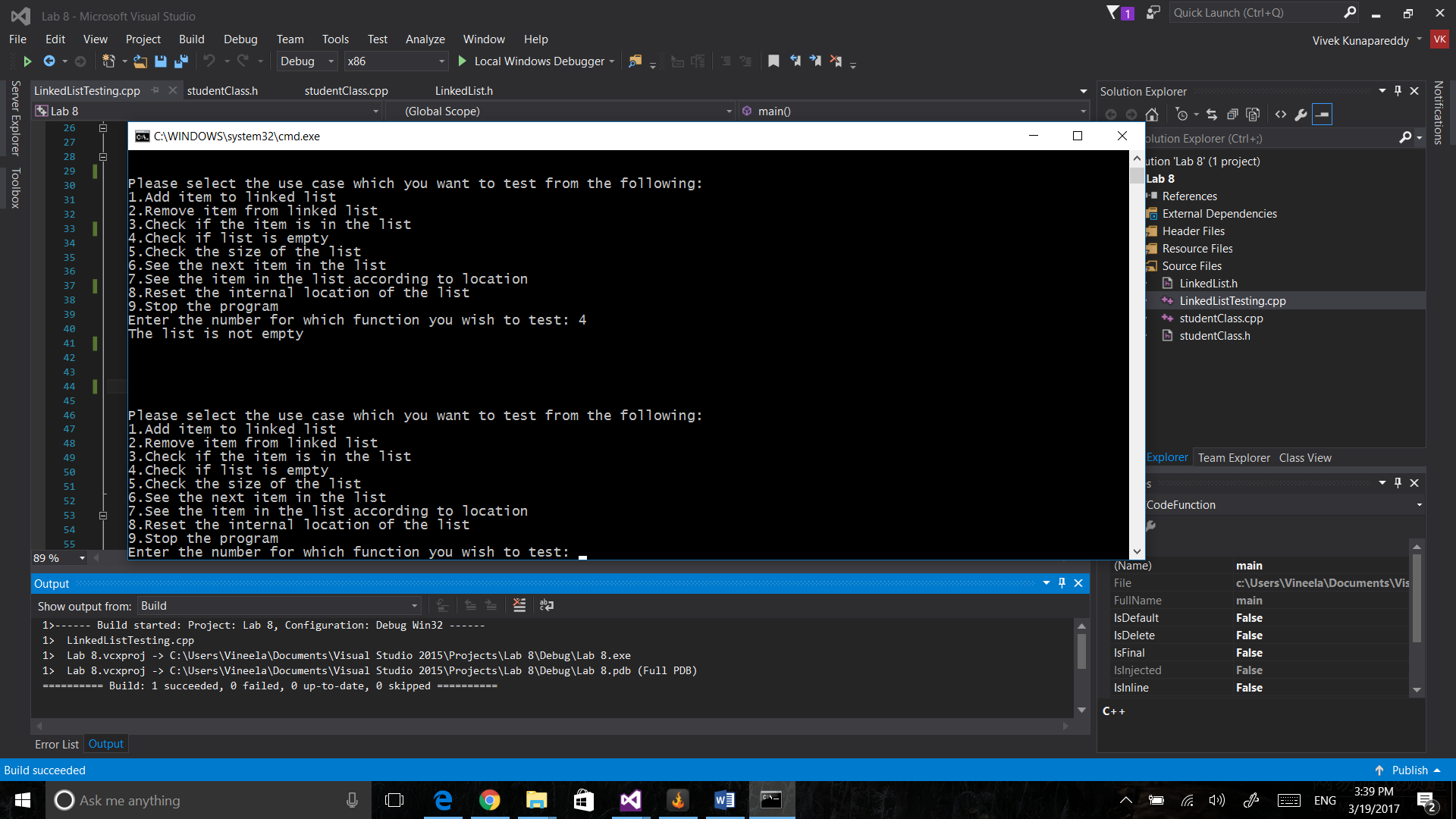
**Screenshots of Task 3:**

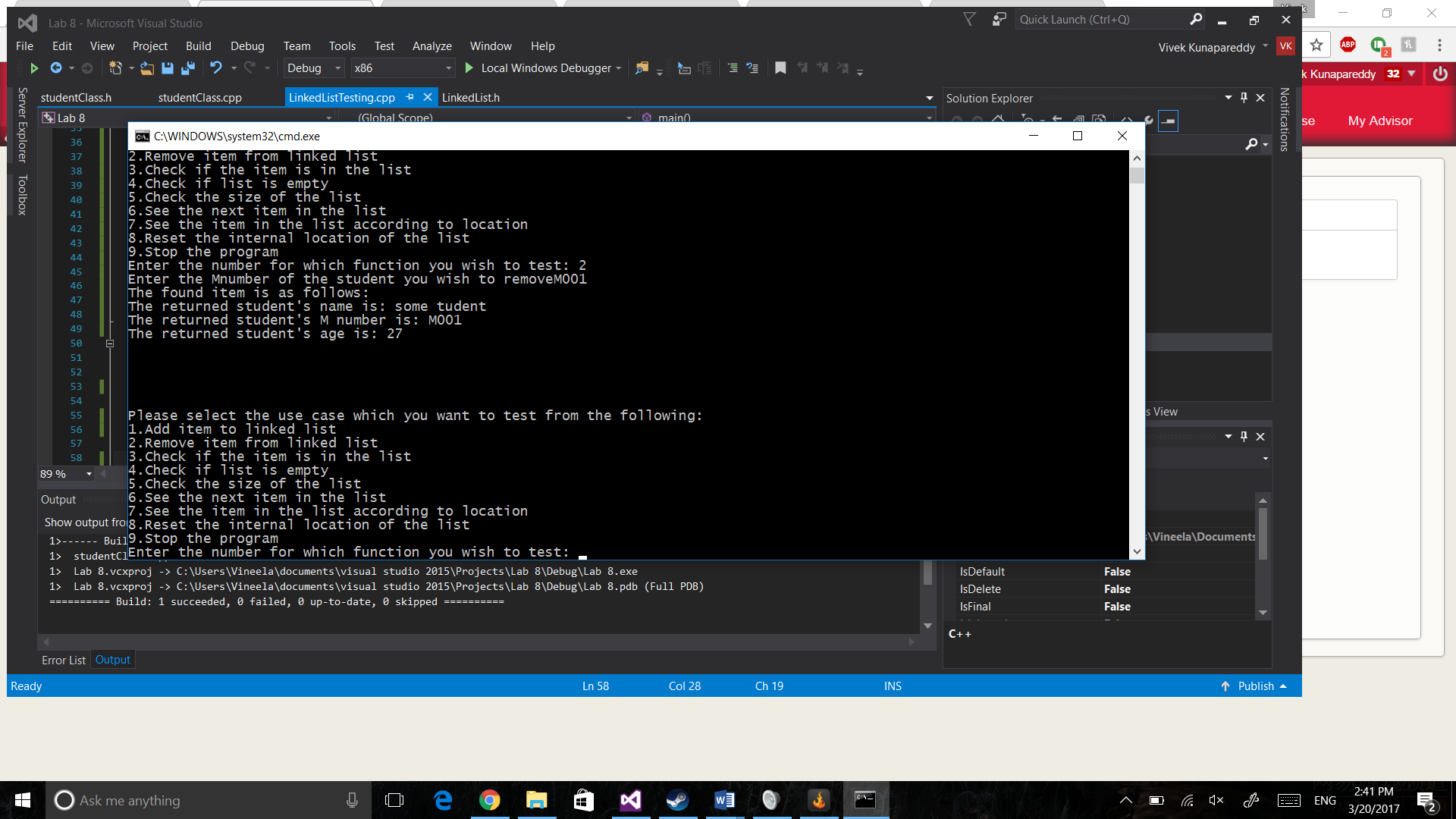
Adding a student to the linked list



Checking size of the list:

Checking if the list is empty:

Removing a student from the list based on M number:

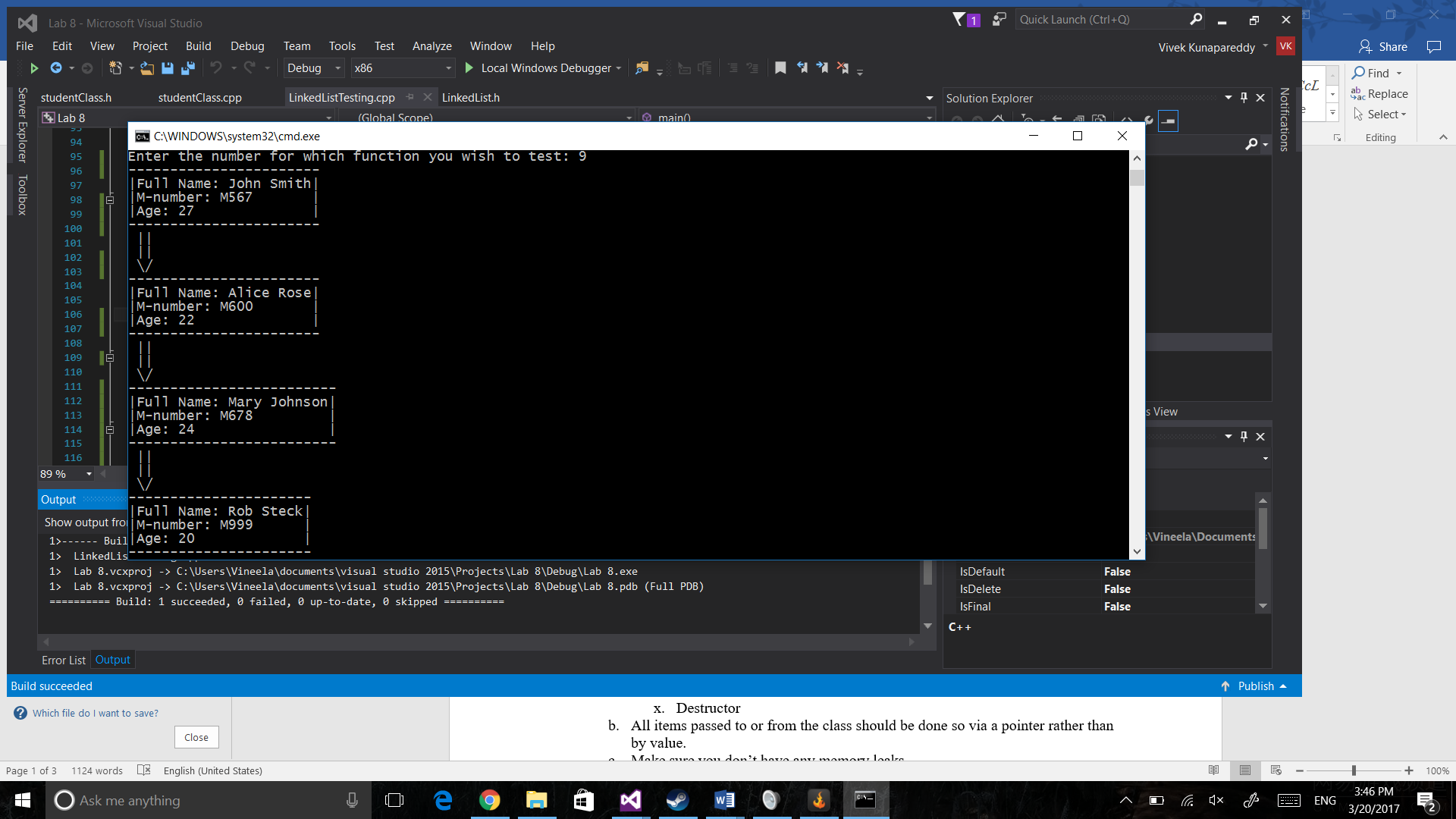


**Task 4 Discussion:**

The members we decided to include in the output are Full Name, M-number, and age of the student.

We decided to ignore GPA as it was an optional field which didn’t need to be inputted. The rest of the fields are vital to identifying a student, hence were included.

**Task 4 Screenshot:**



**Task 4 Discussion:**

The reusable code in Task 4 were the getters defined in the student class as they allowed us an easy way to get access to the private members of the class. None of the code was rewritten for this as LinkedList did not have any code which could be reused. The print function in the LinkedList class uses completely new code which basically uses the student class’ print function

**Instructions to run/compile:**

When entering input for choices, only enter input in the form on numbers from 1-10

**Group contributions:**

The student class was designed by Yuan.

The linked list class was designed by Vivek.

The rest of the assignment was done together.