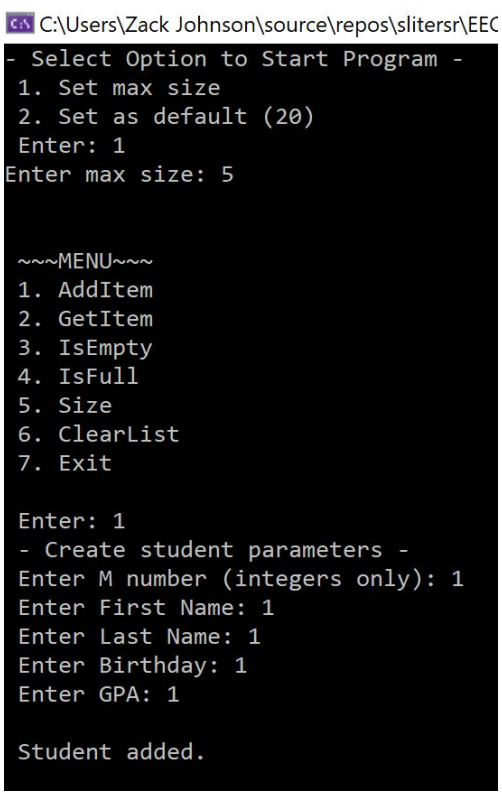
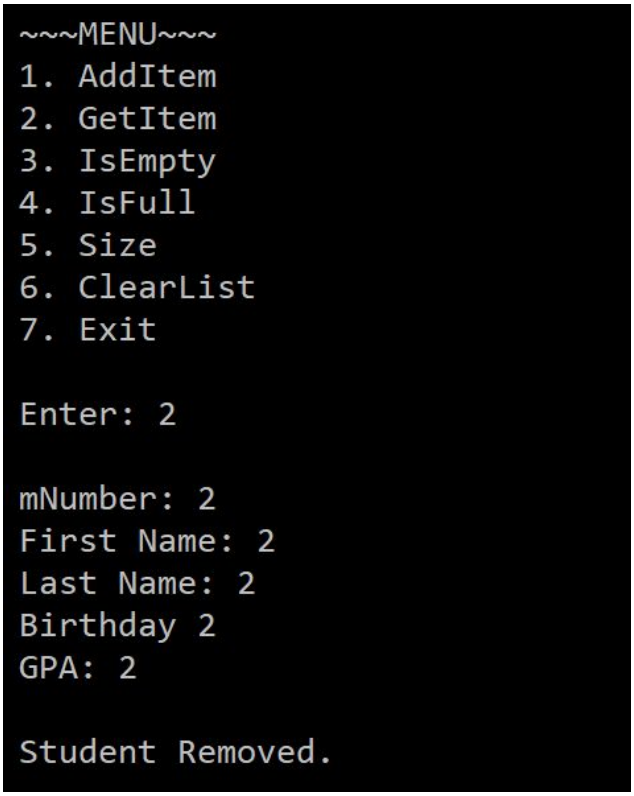
This lab helped us learn the concepts of inheritance, polymorphism, and a continuation of exception handling. We used inheritance to derive the Student class from the Item class and the Stack and Queue class from the Student class. This means that the Stack and Queue class can access the Item's class based on whatever accessor type we give it. Polymorphism was used with our Add Item function which chose between the Stack or Queue AddItem() based on whatever the user chose at the beginning of the program. We also added a few "try catch" blocks to this program to cover any exceptions. These concepts are important for a career in engineering because inheritance is one of foundation concepts to all large code bases. There is a massive amount of code and efficiency is a big thing to think about all the time so having a good inheritance and using polymorphism can make your program easier to use and edit.

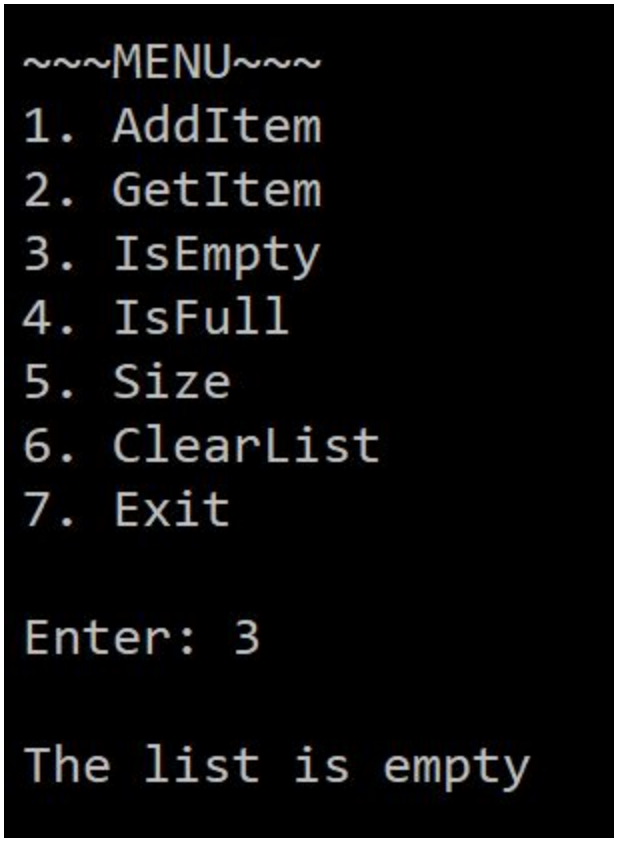
Task 2:



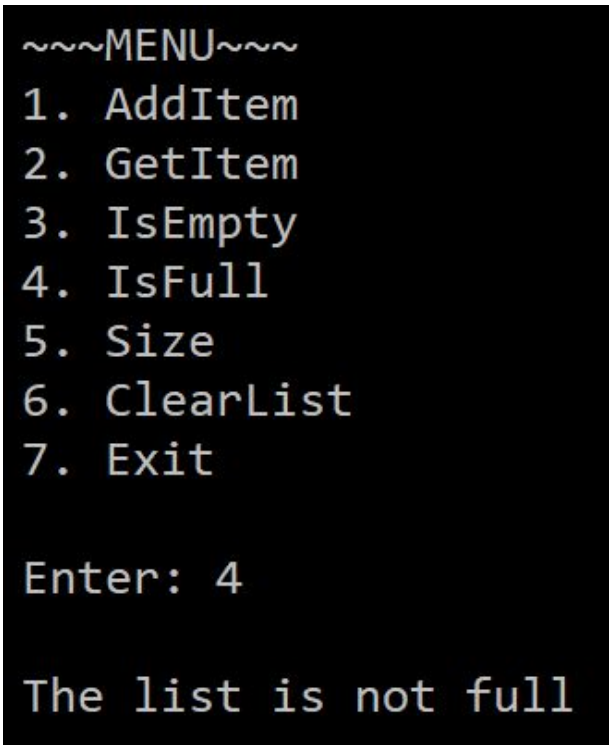
Set Max Size and Add Item



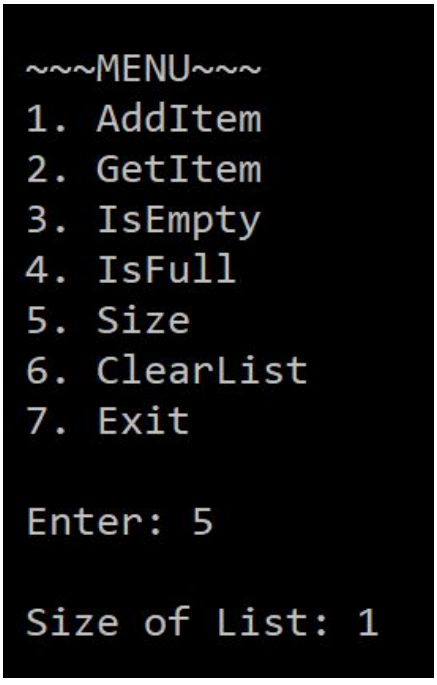
GetItem



IsEmpty



IsFull



ClearList

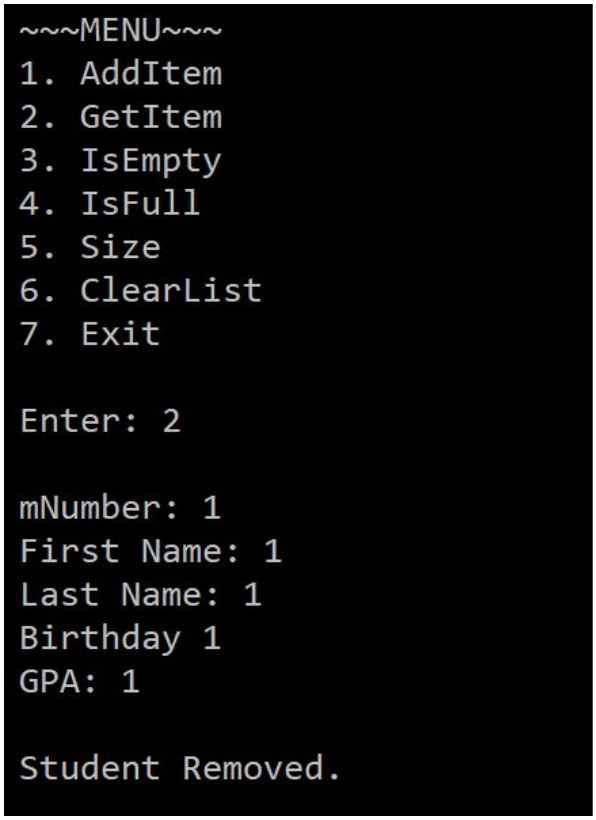
Task 3:

```
- Select Option to Start Program -
 1. Set max size
 2. Set as default (20)
 Enter: 1
Enter max size: 5

- Select Type of List -
 1. Stack
 2. Queue
 Enter: 1

 Creating a Stack...

 ~~~MENU~~~
 1. AddItem
 2. GetItem
 3. IsEmpty
 4. IsFull
 5. Size
 6. ClearList
 7. Exit

 Enter:
```

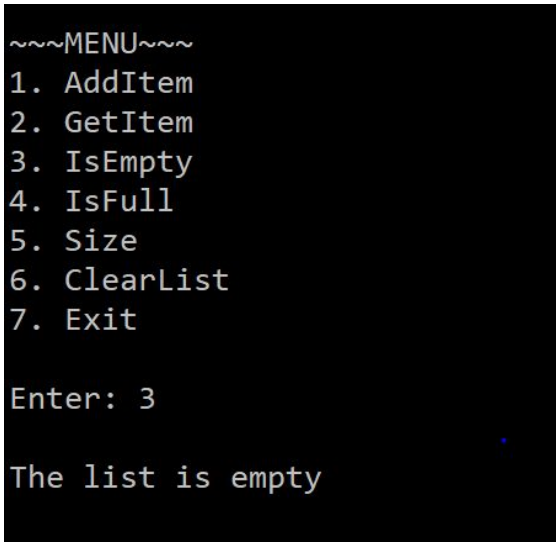
Set Max Size and Create Stack/Queue

```
~~~MENU~~~
1. AddItem
2. GetItem
3. IsEmpty
4. IsFull
5. Size
6. ClearList
7. Exit

Enter: 1
- Create student parameters -
Enter M number (integers only): 1
Enter First Name: 1
Enter Last Name: 1
Enter Birthday: 1
Enter GPA: 1

Student Added.
```

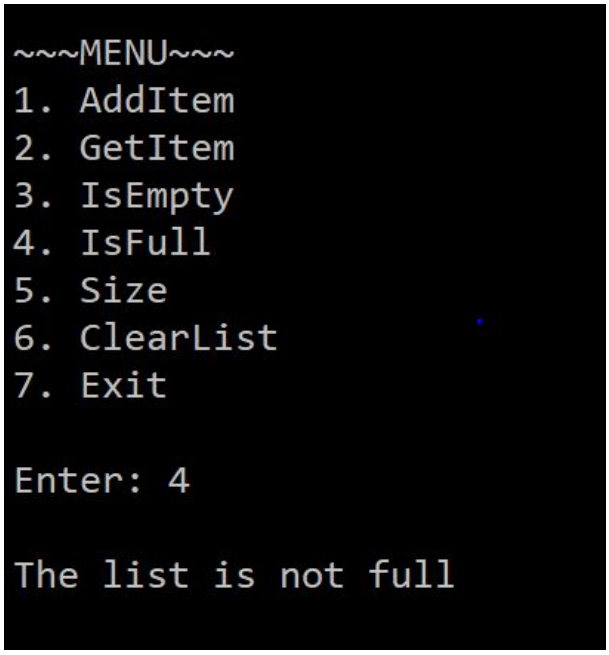
AddItem



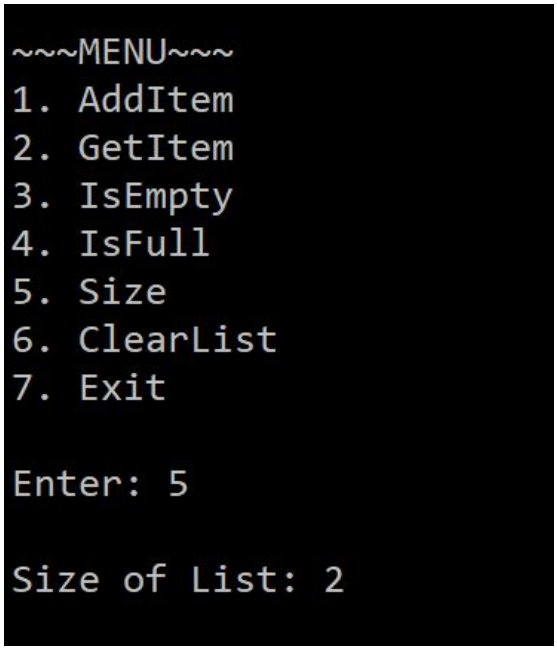
GetItem



IsEmpty

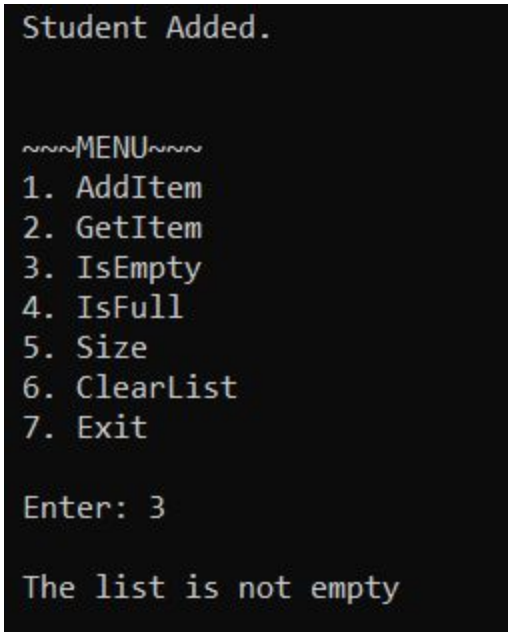


IsFull

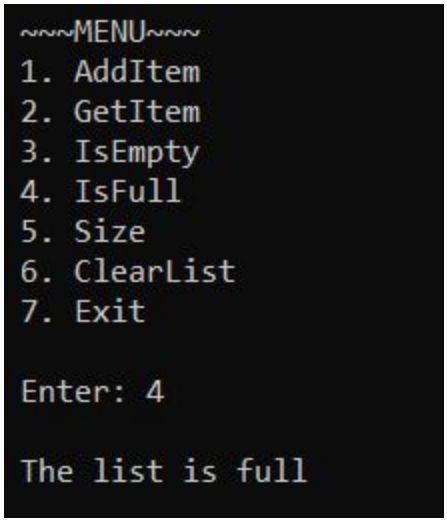


Size


Enter: 6
List has been cleared.
~~~MENU~~~
1. AddItem
2. GetItem
3. IsEmpty
4. IsFull
5. Size
6. ClearList
7. Exit
Enter: 5
Size of List: 0

ClearList
~~~



isEmpty with data in the Stack/Queue



IsFull with data in the Stack/Queue