Wilson Quilli

Professor Elangovan

CMPSC 412: Data Structures Lab

September 9th, 2025

Lab 2: Arrays

In this lab, I implemented different functions in a Vector class using the characteristics of arrays in Python. Python doesn't have a built-in array data structure, but lists act similarly to arrays. To experiment with an array, I created a Vector class that resizes, stores elements, and supports various operations. The *first* function was to check the length of the Vector. I just used the self parameter with the function and returned the self.size, the number of items in my Vector. The *second* function was to check if a certain item was in the Vector. I used the self parameter again and an item parameter. I looped through the Vector size first, then checked if the item was in the Vector using an if statement, and depending on the result, it'll either return True or False. The *third* function was to return an item stored in a specific index of the Vector. First, I used an if statement to ensure the inputted index was within the range and return the element at the index. I used an else statement to raise an IndexError to ensure users remain in the index range. The *fourth* function sets the value at the specified index. The setItem() function has the same structure and code except for the body of the if statement. Inside the if statement, it sets the element at the index. The *fifth* function doubles the capacity of the array length. It goes through the existing array, then copies all the elements, but now with the larger double capacity size. The *sixth* function is the addItem() function that adds a value to the Vector. First, if needed, it resizes

the array, then adds the item to the end of the Vector, and increases the size by 1. The *seventh* function is the insertItem() that inserts a value in the index position. It begins by checking the index and raising an IndexError if needed. Then it checks if a resize is needed. After that, it loops through the Vector and shifts the elements to the right before inserting the new item, if needed. The *eighth* function is the removeItem() that deletes a value from a given index. It stores the item that's being removed in a removed_item variable, shifts all elements to the left, clears the last position, and decreases the size by 1. The *ninth* function is indexOfItem() that returns the index where an item is stored. It loops through the Vector until it finds the item, and if not found, it raises a ValueError saying that the item is not in the Vector. The *tenth* function is extend(), which appends all elements of another Vector into the current one. It loops through the second Vector and adds each item into the first Vector by calling addItem(). The *last* function is subVector() that returns a new Vector made up of elements between two given indices. It first validates the indices, then loops through the range, appending each item to a new Vector. Finally, it returns this new Vector as the subsequence.

Conclusion:

Throughout this lab exercise, I learned how arrays work by implementing a Vector class. I learned how resizing happens when an array reaches capacity, how elements are shifted during insert and remove operations, and how error handling ensures proper index use, so it doesn't just crash, and gives a reason for the error. This lab exercise gave me a better understanding of arrays, how they work, and how similar they are to lists in Python.

Screenshot:

```
v.insertItem(1, 5)
print("Items after insert at index 1:", [v
    .getItem(i) for i in range(v.length())])
removed = v.removeItem(2)
print("Removed item:", removed)
print("Items after remove:", [v.getItem(i) for
    i in range(v.length())])
index = v.indexOfItem(3)
print("Index of 3:", index)
v2 = Vector()
v2.addItem(7)
v2.addItem(8)
v.extend(v2)
print([v.getItem(i) for i in range(v.length
    ())])
sub = v.subVector(1, 3)
print([sub.getItem(i) for i in range(sub
    .length())])
```

```
Length: 3
Contains 2? True
Item at index 1: 2
Item at index 1 after set: 4
Items after insert at index 1: [1, 5, 4, 3]
Removed item: 4
Items after remove: [1, 5, 3]
Index of 3: 2
[1, 5, 3, 7, 8]
[5, 3, 7]

=== Code Execution Successful ===
```

Zoom Link:

https://psu.zoom.us/rec/share/K1qB4eRx26Xz3qqsnJvwHUNXgatotmMJ-wvcXxKwg4zciQl68yl-WjQwIniFBC8s.iu0XMiZ2729cxg6O