Wilson Quilli

Professor Elangovan

CMPSC 412: Data Structures Lab

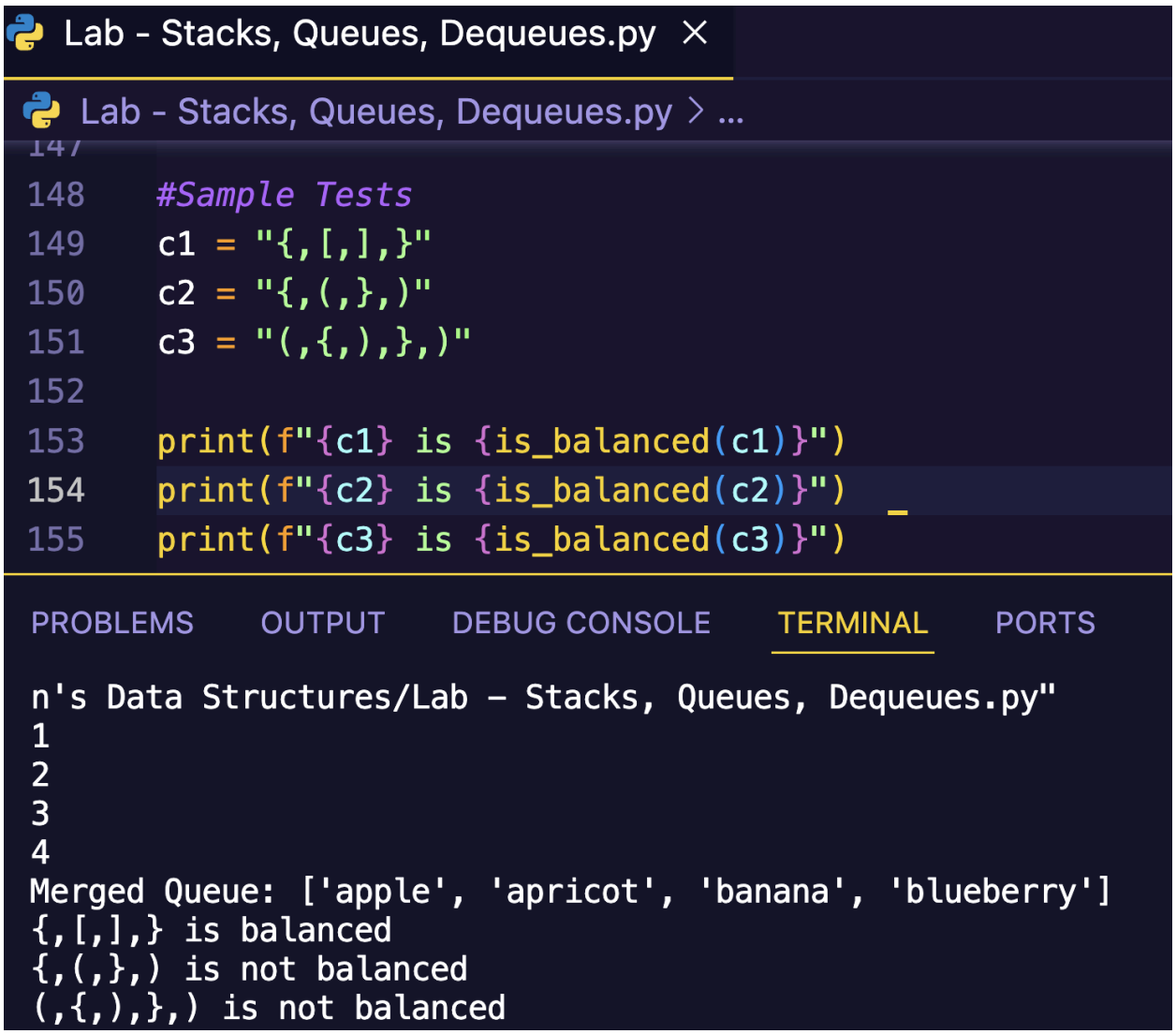
September 15th, 2025

# Lab #3 - Stacks, Queues, and Dequeues

In this lab, I practiced three exercises, implementing Stacks and Queues data structures. The first exercise I completed was implementing a Queue using two Stacks. This was done using one Stack as an input Stack for the Queue, enqueuing new elements by pushing them. I used my second Stack as an Output Stack for the Queue, dequeuing elements by popping them. The first Stack just pushes elements in an O(1) enqueue operation. These two Stacks cooperate together to implement a Queue following the First-in-First-Out rule. In the second exercise, I implemented a Queue class containing enqueue, dequeue, and isEmpty functions. Then, I used a function to merge two queues together into a third queue. This merging process iterates through both queues, comparing elements and enqueuing, with a O(n + m) time complexity, where n and m = sizes of the queues. In the third exercise, I implemented a Stack class containing isEmpty, Push, and Pop functions to check if characters such as () and {} have a closing partner or are balanced. I did this by using a new Stack Object, a dictionary to store each left-right character. Then using the Stack, I iterated through it using a For Loop. Whenever a left bracket was found, it was pushed into a Stack and for the right bracket, the top of the Stack was popped and compared, ensuring each left bracket has a right bracket. The time and space complexity for this exercise is O(n), since it’s

searching and comparing each value in the Stack. Finally, using the test cases that were provided, I tested out the function and checked which were balanced and which weren't.

**Screenshots:**



# Resources


- GeeksforGeeks. (2025, September 13). *Stack in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/python/stack-in-python/
- W3Schools. (n.d.). *Python – Data Structures – Queues*. W3Schools. https://www.w3schools.com/python/python_dsa_queues.asp