

Credit: CSE3110 - Iterative Algorithm 1  
Assignment: QueueList

How has your code changed from planning to coding? Please explain.

Because I had coded methods for the linked list during the StackList mastery I already had the methods I needed to complete the mastery. The Stacklist mastery prior to this also deepened my understanding on how these iterative algorithms worked so that made coding this mastery much easier. My approach to this mastery didn't change from my original plan; however, I did make some of the same mistakes I did during StackList.



The screenshot shows an Eclipse IDE window with a Java application named 'QueueListTesting'. The code in the editor is as follows:

```
2
3 import Queue2And3_SkillBuilders.Queue3;
4
5 public class QueueListTesting {
6
7     public static void main(String[] args)
8     {
9
10         QueueList q = new QueueList();
11
12         System.out.println("Adding '1', 'red', and 'bob123' to queue." + '\n');
13
14         q.enqueue("1");
15         q.enqueue("red");
16         q.enqueue("bob123");
17
18         System.out.println("Front of queue: " + q.front());
19     }
20 }
```

The console output shows the message: "Adding '1', 'red', and 'bob123' to queue." followed by a newline. Below this, a red error message is displayed:

```
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "LinkedListParts123_SkillBuilders.Node.getNext()" because "current" is null
    at Chapter13/LinkedListParts123_SkillBuilders.LinkedList.addAtEnd(LinkedList.java:91)
    at Chapter13/Mastery.QueueList.enqueue(QueueList.java:53)
    at Chapter13/Mastery.QueueListTesting.main(QueueListTesting.java:14)
```

```
public void enqueue(Object item)
{
    String s = "" + item;

    if (isEmpty())
    {
        rear = 0;
        front = 0;
        list.addAtEnd(s);
    }
    else
    {
        rear += 1;
        list.addAtEnd(s);
    }
}
```

Trying to add items to the end of a linked list when it's empty doesn't work due to how the method is coded. Instead I had to add to the front of the linked list. Upon reflection, I could've just made the enqueue method only use the addAtEnd LinkedList method if I had changed the LinkedList addAtEnd method to use addAtFront if the list was empty.