Implementing Iterators *Design Considerations*

- Possible approaches
  - Define iterator functionality as ADT operations (write member methods for hasNext(), next(), and remove())
  - Implement the iterator methods within their own class
    - Subclass (array or linked implementation)
    - Independent class (possibly less efficient)

Iterator
Member Methods

- Disadvantages to having methods hasNext() and next() in the ADT
  - Only one iterators at a time
  - Interface bloat

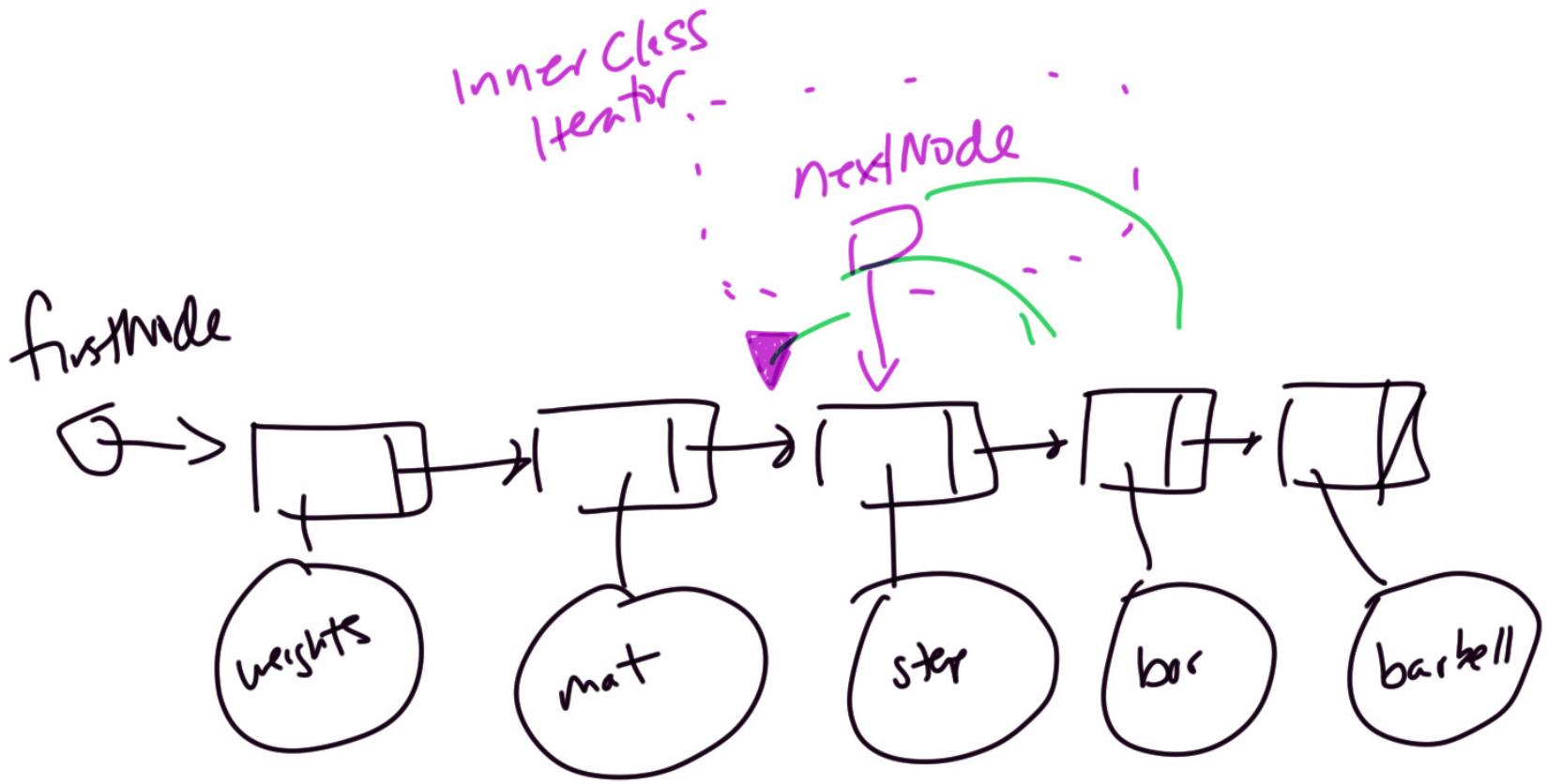
Iterator Implementation Approaches

1) separate class iterator (independent of implementation)
2) inner class iterator for linked chain implementation
3) inner class iterator for array implementation

Iterator with no knowledge of list's implementation

→ variable for the list

→ variable for the position of the iterator (ex: 3)



Next
→ check hasNext() AKA nextNode != null
→ store nextNode.data to return
→ move nextNode to nextNode.next

* The list should return an instance of the iterator to implement Iterable