• a description of the design of your doubly-linked list implementation. (A couple of sentences will probably suffice, perhaps with a picture of a typical Sequence and an empty Sequence. Is the list circular? Does it have a dummy node? What's in your list nodes?

• Each node has a value and next and previous pointing

• Only set the head node, doesn’t have tail node

• Not circular

• pseudocode for non-trivial algorithms (e.g., Sequence::remove and interleave).

• a list of test cases that would thoroughly test the functions. Be sure to indicate the purpose of the tests. For example, here's the beginning of a presentation in the form of code:  
The tests were performed on a sequence of strings (i.e., the ItemType typedef specified std::string).  
 // default constructor

• Sequence s;

• // For an empty sequence:

• assert(s.size() == 0); // test size

• assert(s.empty()); // test empty

• assert(s.remove("paratha") == 0); // nothing to remove

•   
  
Even if you do not correctly implement all the functions, you must still list test cases that would test them. Don't lose points by thinking "Well, I didn't implement this function, so I won't bother saying how I would have tested it if I *had* implemented it."