Singly Linked Lists   *(find and work with a partner)*

1. If everything were working correctly in our `SinglyLinkedList` class, what would be printed after executing the following code in `main`?

```
public class CourseManager {

    public static void main(String[] args) {
        SinglyLinkedList<String> courses = new SinglyLinkedList<String>();
        courses.addLast("Math 102");
        courses.addFirst("CS 106");
        courses.addFirst("CS 105");

        System.out.println("Number of courses: " + courses.size());
        System.out.println(courses);
    }
}
```

2. Our `SinglyLinkedList` class from last time began like this:

```
public class SinglyLinkedList<E> {

    private Node<E> head; // pointer to the beginning of the list
    private int size;     // number of elements in the list

    public SinglyLinkedList() {
        head = null;      // null is a common choice here
        size = 0;         // no elements yet
    }
```

Fill in the rest of the `addFirst` method below (draw a diagram). What is the runtime of this method?

```
public void addFirst(E newData) {
    Node<E> newNode = new Node<E>(newData);




}
```

3. What is the runtime of the `addLast` method?

4. If I wrote a `get` method that took in an integer `i` and returned the data in the `i`th `Node`, what would be the runtime of this method?
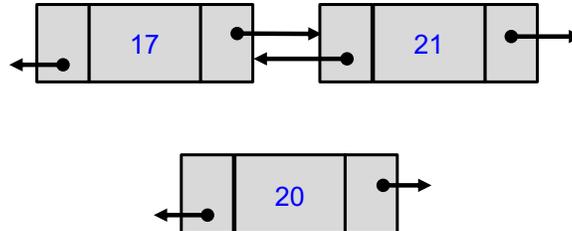
5. Fix the error(s) in the code for `addLast` below:

```
public void addLast(E newData) {
    Node<E> newNode = new Node<E>(newData);
    if (size == 0) {
        head = newNode;

    } else {

        Node<E> currNode = head;
        while (currNode != null) {
            currNode = currNode.next();
        }
        currNode.setNext(newNode);
    }
    size += 1;
}
```

Doubly Linked Lists

1. In the digram below, I want to insert the `Node` with data 20 in between 17 and 21.

(a) Label the `Node`s we refer to as `curr` and `new`.

(b) Write out pseudocode (4 lines) to show all the pointer updates that must happen to insert the new `Node`.