CMPSCI 187, Spring 2015 Discussion #9
Doubly-Linked Lists: Group Response Sheet

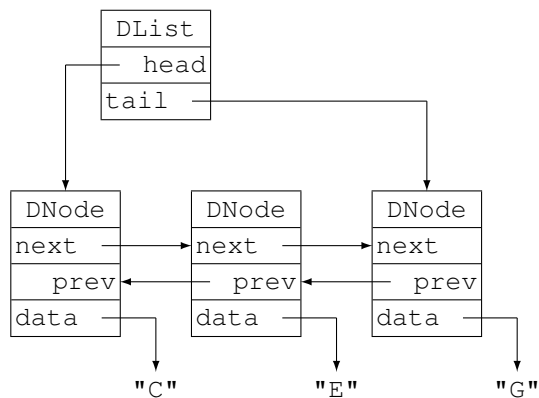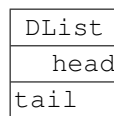Marc Liberatore and John Ridgway
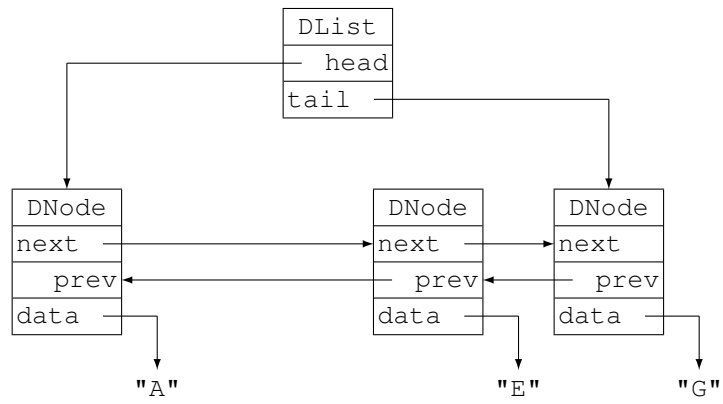
30 March 2015

Names: _____

Each *pair* of students should hand in one response sheet.

Question 1: To satisfy yourself that this method always works draw a picture that shows adding a node to an empty list; a second that shows adding a node to the beginning of a non-empty list; a third that shows adding a node to the end of a non-empty list; and a fourth that shows adding an element in the middle of a list.

```
DList
     head
tail
```

```
DList
     head
tail
```

```
DNode        DNode        DNode
next         next         next
  prev         prev         prev
data         data         data

 "C"          "E"          "G"
```

```
                         ┌──────────┐
                         │  DList   │
                         ├──────────┤
                         │   head   ├──┐
                         ├──────────┤  │
                 ┌───────┤   tail   │  │
                 │       └──────────┘  │
                 │                     │
                 │                     │
   ┌──────────┐  │   ┌──────────┐   ┌──────────┐
   │  DNode   │  │   │  DNode   │   │  DNode   │
   ├──────────┤  │   ├──────────┤   ├──────────┤
   │ next     ├──────│ next     ├───│ next     │
   ├──────────┤  │   ├──────────┤   ├──────────┤
   │     prev │◄─────│     prev │◄──│     prev │
   ├──────────┤      ├──────────┤   ├──────────┤
   │ data     ┤      │ data     ┤   │ data     ┤
   └──────────┘      └──────────┘   └──────────┘
         │                 │              │
        "A"               "E"            "G"
```

Question 2: Write a method `void` `removeNode(DNode<T> nodeToRemove)` that takes a reference to a particular node in the list and removes that node.

(Question 3 is on the other side.)

2

Question 3: Write a method `moveInto` with the following signature:

```
void moveInto(DList<T> other, DNode<T> insertAfter)
```

The method moves all of the elements from `other` into the current list, after the element `insertAfter`. If `insertAfter` is **null**, they should be inserted at the start of the list. Note that the `other` list should end up empty.