Version 2

- Let’s re-think `NoteArray` as a collection of *elements* that we can shuffle around as we’d like.
- We can make any element follow any other element.
- Call each element a *node*. 
What’s in each element?

**NoteNode**

### Data
- it’s Note
- what comes next
- another NoteNode

### Actions
- getPhrase()
- get the phrase beginning at this Note and continuing until the end

Introducing the **Linked List**

- A linked list is information broken into smaller pieces, where each piece knows the next piece, but none other.

Version 2: **Note Linked List**

```java
class NoteNode {
    //---/---/ Fields //---/---/
    Note note;
    NoteNode next;

    //---/---/ Constructors //---/---/
    public NoteNode(Note note) {
        this.note = note;
    }
}
```

Version 2: **Note Node**

No Access Modifier creates package-private members
- Allows direct access from within package
//---//----// Methods //---//----//

public Note getNote()
{
    return note;
}

public void setNote(Note note)
{
    this.note = note;
}

public static void main(String[] args)
{
    NoteNode n1 = new NoteNode(new Note(JMC.D2, JMC.ENT));
    NoteNode n2 = new NoteNode(new Note(JMC.FS2, JMC.ENT));
    NoteNode n3 = new NoteNode(new Note(JMC.A2, JMC.ENT));
    n1.next = n2;
    n2.next = n3;
}

public NoteNode getNext()
{
    return next;
}

public void setNext(NoteNode next)
{
    this.next = next;
}

public static void main(String[] args)
{
    NoteNode n1 = new NoteNode(new Note(JMC.D2, JMC.ENT));
    NoteNode n2 = new NoteNode(new Note(JMC.FS2, JMC.ENT));
    NoteNode n3 = new NoteNode(new Note(JMC.A2, JMC.ENT));
    n1.setNext(n2);
    n2.setNext(n3);
}
public int size()
{
    NoteNode current = this;
    int size = 0;
    while(current != null)
    {
        size++;
        current = current.next;
    }
    return size;
}

public Phrase getPhrase()
{
    NoteNode current = this;
    Phrase phr = new Phrase();
    while(current != null)
    {
        phr.addNote(current.note);
        current = current.next;
    }
    return phr;
}

public static void main(String[] args)
{
    NoteNode n1 = new NoteNode(new Note(JMC.D2, JMC.ENT));
    NoteNode n2 = new NoteNode(new Note(JMC.FS2, JMC.ENT));
    NoteNode n3 = new NoteNode(new Note(JMC.A2, JMC.ENT));
    n1.next = n2;
    n2.next = n3;
    Phrase phr1 = n1.getPhrase();
    Part part1 = new Part("Bass", JMC.PBASS, phr1);
    Score score = new Score("Buster", 100.0, part1);
    View.notate(score);
}
After the following code is run, if we begin traversing the linked list at node4, what is the order of the list?

node2.setNext(node1);
node4.setNext(node2);
node3.setNext(node1);
nоде1.setNext(node5);

A. 1 2 3 4
B. 4 3 2 1
C. 4 2 3 1 5
D. 4 2 1 5
E. 5 1 2 3 4