Recitation Guide for February 18, 2008

I. Housing Keeping
    a. Homework 4 – Due Saturday February 23rd 7pm.

    b. Drop day – Friday February 29th.

    c. Pair-programmers' agreement – Due Wednesday March 5th in class (same day as Quiz 2)

II. Pair – Programming:
    a. General overview
        i. Driver – types in the code at the computer
        ii. Navigator – reviews the lines of code as it is typed in
        iii. More resources:
            1. http://coweb.cc.gatech.edu/cs1316/188
            2. http://en.wikipedia.org/wiki/Pair_programming

    b. Pair-programming assignments start with homework 6.

    c. Pairs must turn in a pair-programmer's agreement and post the pair to the coweb pairs page (http://coweb.cc.gatech.edu/cs1316/1035) to have the assignment graded.

III. Linked lists, JMusic Theory and homework 4
    a. Review of `static` methods and variables
        i. `static` methods and variables are accessible without declaring an instance of the class that houses them. As a side note, all methods in `SongPhrase` are `static`.
        ii. Examples:
            1. `Math.cos(double number)`
            2. `FileChooser.setMediaPath(String mediaPath)`
            3. `SongPhrase.AG1()`

    b. Phase data
        i. Comes in pairs: First is the actual note (i.e. `JMC.G4`) and then the duration (i.e. `JMC.QN`).
        ii. Example analyzed

```
1  public static Phrase riff1() {
2     double[] phrasedata = {JMC.G3, JMC.EN,
3     JMC.B3, JMC.EN, JMC.C4,JMC.EN, JMC.D4, JMC.EN};
4     Phrase myPhrase = new Phrase();
5     myPhrase.addNoteList(phrasedata);
6     return myPhrase;
7  }
```

Line 1: `riff1` method: `public static` method that returns a `Phrase` and takes in no parameters.

Line 2 -3: an array of `double` values called `phrasedata` is declared and initialized with an array of `double` values{ G3, eighth note, B3, eighth note, C4, eighth note, D4 eighth note}.

Line 4: A `Phrase myPhrase` is declared and initialized to an empty `Phrase`

Line 5: The array of `double` values is added to `myPhrase`

Line 6: `myPhrase` is returned.

c. JMusic Theory
   i. In general:
      1. A `Score` contain `Part`s, a tempo, a time signature,
      2. A `Part` contain `Phrase`s and an instrument
      3. A `Phrase` contains the musical data composed of an array of notes and their durations

d. Homework 4
   i. The linked list – `AdvancedSongList`
   ii. The linked list node – `AdvancedSongNode`
   iii. The data within each linked list node – `Phrase`
   iv. Everything is very similar to what you did for homework 3 only this time you do not have to write the linked list methods.
   v. New methods in `AdancedSongNode`:
      1. `public AdvancedSongNode copy()`
   vi. New methods in `AdvancedSongList`:
      1. `public void repeatNext(AdvancedSongNode afterThisNode, AdvancedSongNode newNode, int count)`
      2. `public void repeatNextInserting(AdvancedSongNode afterThisNode, AdvancedSongNode newNode, int count)`
      3. `public void showFromMeOn(int instrument)`
      4. `public void weave(AdvancedSongNode afterThisNode, AdvancedSongNode newNode, int count, int skipAmount)`
   vii. Minor note: You can only get 2 `Part`s on a `Score` before it becomes necessary to set the channel or else any new `Part`s will overwrite old ones. Assume we already have some `myScore` declared and set up previously then all you have to do to set the channel for each `Part` is:
      ```
      for (int i = 0; i < myScore.getPartArray().length; i++){
            myScore.getPart(i).setChannel(i);
      }
      ```

e. Questions?