21M.269 Spring 2012
Assignment 2 : Problem Set 1

Implement the following without using `music21` (use any programming language you're comfortable with for 1–3, but you might as well get familiar with python since you need it for #4.)

Readings: Tymoczko Ch. 2 (2.4 in particular, 2.10 for tuning discussion) (35 pages)

1) Write a short script that converts a frequency (in Hz) to a numerical pitch label, as discussed in Tymoczko 2.1 (numerical pitch labels are also known as midi pitch numbers). Assume standard Western equal temperament tuning.

2) Write a script that, given a list of numerical pitch labels, transposes them by an interval given as an integer representing the number of semitones (for instance, 1 = one semitone (minor second), 2 = two semitones (major second), 3 = three semitones (a minor third), etc).

3) Write a script that determines the interval between any two notes. Be sure the script can accept different input formats letter name with octave (i.e. 'C4') and numerical pitch labels. Your code should be able to return the interval in a variety of formats (as specified by an additional input) including the number of semitones and the interval name as a string. Be sure your script is robust to enharmonic differences and the order of the notes.

4) Orchestras are often thought of as divided into sections based upon the type of instrument contained within those sections (woodwinds, brass, strings, etc). These types of instruments all have their own unique properties, but within these sections, each instrument itself has specific characteristics, such as a transposition, a range of notes within which it is feasible for a performer to play, and a variety of extended techniques available to the performer. On the OCW site you find a python file outlining a class hierarchy (`instrument.py`) containing the parent class, `Instrument()`, and two subclasses, `String()` and `Woodwind()`. Each of these subclasses have their own subclasses, representing instruments that belong to them (`Violin()`, `Cello()`, `Clarinet_BFlat()`, etc.) as well as attributes (`transposition`, `range`, `extendedTechniques`).

    a) Write two more subclasses of `Instrument()` on the same level as `String()` and `Woodwind()`, and write three subclasses for each of those (on the same level as `Violin()` and `Clarinet_BFlat()`). (although, see if you can think of subclasses of Instrument() that have another level of subclasses before the class level that involves specific instruments)

i. For each class, think about which attributes can be inherited, and which will need to be overridden. Fill in the default argument values in the `__init__` method for each.

b) Write the `convertSoundingRangeToWritten()` method under the parent `Instrument()` class. Demonstrate that your code works by filling in the correct output for that method's doctest and ensuring that it runs without errors.

c) Historically, different ensembles have come to be associated with stereotypical instrumentations (for instance, a string quartet generally contains two violins, a viola, and a cello). Open a new module and write an `Ensemble()` parent class, with three different subclasses representing types of ensembles. Create attributes such as size, standard instrumentation, and a short repertoire list.
   i. Update the instrument classes with a method that takes in an `Ensemble()` object as input and returns a boolean stating whether or not the instrument is part of that ensemble.
   ii. Add a method to `Ensemble()` that allows you to add `Instrument()` objects, and overwrite this method in the subclasses of `Ensemble()` as the need arises.

Extended questions: Choose one of the following questions to be worked on in a group of two or three people. One piece of code may be submitted per group, but each group member must submit a brief written explanation of what they personally contributed to the project. All code must have thorough documentation for all classes and methods, as well as working doctests that prove the code's functionality and robustness.

A) Create a class hierarchy of "noise makers" (literally, anything that makes sound…cars, pigs, violas, your roommate…) and think of attributes and methods that they each might have (particularly, think about which ones can be inherited).

B) Write a script that converts a list of numerical pitch labels in Western equal temperament to frequencies of
   i) just intonation
   ii) Pythagorean tuning

   Assume the note C4 (60 in midi number notation) to be the basis for your calculations.

   (feel free to refer to Wikipedia for more information on different tuning systems)

21M.269 Studies in Western Music History: Quantitative and Computational Approaches to Music History

Spring 2012