

CS 1316 - Homework 6 – Music and Sound

Due before 6pm on Monday October 12th.

This is an individual assignment. You may collaborate with other students in the class but your solutions must be your own.

For this assignment, you will be creating a subclass to Sound called MusicSound which adds some basic music functionality. MusicSound will have two static methods which are used to produce musical tones.

Note: Sound does not have a default constructor, in order to instantiate MusicSound you need to override a constructor using a parameter that is understood in the superclass.

Sine Wave

In order to produce normal sounding tones, we need a static method to make sinusoidal waves. This method should take two parameters for time in seconds (double) and samples per wavelength (int). An example syntax:

```
MusicSound sWave = MusicSound.sin(0.3,100);
```

For further details see: http://en.wikipedia.org/wiki/Sine_wave

Remember that Frequency

is measured in Hz, or wavelengths per second. With CD quality sound, each second of sound has 44,100 samples. The SimpleSound class (which is sub-classed by the Sound class) uses a default sampling rate of $\frac{1}{2}$ of this, or 22,050 samples per second.

Notes

Now, using our sin wave function we can dial in frequencies to produce specific tones. The method to do this should take in three parameters for time (double), pitch (String), and octave (int).

The frequencies for specific pitches are constants; for the scope of this assignment you only *need* to implement three tones:

- “C”: 16.35 hz
- “E”: 20.60 hz
- “G”: 24.50 hz

- Extra Credit - add all 12 tones: <http://www.phy.mtu.edu/~suits/notefreqs.html>

Your code should parse the input string to set the frequency; bad input should default to C (16.35). The octave should double the frequency at each index, such as:

- C0 = 16.35 , C1 = 32.70 , C2 = 65.41 , C3 = 130.81 , etc.

An example syntax:

```
MusicSound c = MusicSound.note(0.3,"C",5);
```

Resources

This homework may call for a bit more prior-knowledge on physics and music, so I'm including some links to relevant information.

- <http://en.wikipedia.org/wiki/Frequency>
- <http://en.wikipedia.org/wiki/Hertz>
- <http://en.wikipedia.org/wiki/Octave>
- http://en.wikipedia.org/wiki/Scientific_pitch_notation
- http://en.wikipedia.org/wiki/Chromatic_scale

Extra Credit

- Build an even more rugged interface for note() with parameters for volume or other effects.
- Provide a main method in MusicSound which constructs a song with your framework and plays it.
- Other interesting functionalities will be considered, just ask Dr. Summet or a TA.

Hints

- Math.sin accepts radians, not degrees! Look at Math.toRadians() for help!
- aString.equals("C") works better than aString == "C"

Grading Criteria

- Sine Wave
 - Header correct 5pt
 - Produces a sine wave 10pt
 - Sine wave is the correct length and frequency 15pt
- Music Notes
 - Header correct 5pt
 - Parses string input correctly 25pt
 - Dials in correct frequency 30pt
 - Sound is the correct length 10pt