Programming Assignment Checklist: Digital Signal Processing

Pair programming. On **THIS** assignment, you are encouraged (not required) to work with a partner provided you practice pair programming. Pair programming "is a practice in which two programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, or test." One partner is driving (designing and typing the code) while the other is navigating (reviewing the work, identifying bugs, and asking questions). The two partners switch roles every 30-40 minutes, and on demand, brainstorm. Before pair programming, you must read the article All I really need to know about pair programming I learned in kindergarten.

You may choose a partner from the same or a different precept. You and your partner will each turn in a **separate individually written readme.txt** for the assignment, stating who your partner is and detailing your experience. However, only one of you should turn in the assignment (the .java files).

Frequently Asked Questions

I'm having trouble with trigonometry. What could be wrong? It's fine to just use the formula provided. Note that a sine wave repeats every 2π radians. Use `Math.PI` to get the mathematical constant π. Recall that in Java, the argument to `Math.sin()` is in radians, not degrees.

I am getting a runtime error when I run EchoFilter with `felten.mp3` or `pearlharbor.mp3`. What should I do? Download `stdplayer.jar` again.

I am getting a warning when I run EchoFilter with `felten.mp3` or `pearlharbor.mp3`. What should I do? This may be ok. Check that bushism.mp3 works without warnings, and that felten.mp3 works until sample 705, and then gets a warning. Warnings before then or in during the playing of bushism.mp3 are a sign that something is wrong with your code.

I can play CDs and MP3s using my favorite media player, but I don't get any sound out of Java. What could be wrong? First, be sure to try the JavaLayer Player `javazoom.jl.player.jlp` as suggested in the assignment - this will ensure that the problem is not in any of your code. If you don't get any sound, it may be because your operating system has turned off the sound device that Java uses. On my Windows XP machine, I had to go to *Start -> Programs -> Accessories -> Multimedia -> Volume Control* and unmute the *Wave Out* slider. The 017 labs in the Friend center may need a similar adjustment.

How do I convert from a `double` to an `int`? Use an explicit cast: `int n = (int) x;` - this will truncate the fractional part of `x`. A slightly better solution is to round to the nearest integer using: `int n = (int) Math.round(x);`. The cast is still needed since `Math.round()` returns a `long`.

How do I determine how many elements are in an array? If you're asking this, you probably need to review Section 1.4.

I get an ArrayOutOfBounds or NullPointerException error. What could cause this? Do your constructors initialize all of the data members (left, right, and N)? Did you allocate memory for each of your arrays with `new`? Did you inadvertently redeclare `int N` or `double[] left` in a method or constructor, thereby hiding the instance variable with the same name?

Do I need to worry about adding two waves that have a different number of samples? Or a wave whose left and right channels have a different number of samples? No. It's fine to assume all waves within a given program will have the same length (but that number need not be 44,100 or 1,152), but it can't hurt to do some error checking.

Should I play the echo of the last 10 Waves, after all the Waves have been read in? Yes, the last Wave read in is played with an echo, but there are 9 remaining echos in your array that have not been played. They should

be played without adding them to a Wave from the mp3 file.

Why do I need to include `System.exit(0)` at the end of my program? There is a documented bug in the Java VM that prevents the program from exiting cleanly when you interact with a sound card. This is the temporary work-around.

Why can't I just put `StdPlayer.java` in my working directory and avoid the classpath stuff as with `StdDraw.java`? You can, but `StdPlayer.java` relies on 78 other user-defined files, so you'd have to put all of them in the same place too.

The first wave I plot is incredibly slow. How can I fix this? Be sure to turn animation mode on before you plot the first wave using `StdDraw.show(0)`. Then, after plotting each wave, redisplay using `StdDraw.show(10)`.

Java complains that it can't find `javazoom.jl.player.jlp` even though `stdplayer.jar` is in the current directory. What am I doing wrong? It should be `javazoom.jl.player.jlp` with the letter `l` and not the number `1`. There is a lesson to be learned here about selecting good variable names.

Input, Output, and Testing

Input. Here are some sample input files.

Compilation. Your programs must be named `Wave.java`, `EchoFilter.java`, and `MP3Viewer.java`. Don't forget to hit the "Run Script" button on the submission system to test that it compiles cleanly.

Submission

Use the following readme file template.

Possible Progress Steps

These are purely suggestions for how you might make progress. You do not have to follow these steps.

- Download the directory `dsp` to your system. It contains the MP3 Player library, and some sample client programs.
- Read Sections 3.1 and 3.2. Review the data type Vector.java (Program 3.2.x) for N-dimensional real vectors. It is similar in many ways to `Wave.java`.
- We recommend defining the set of values as follows:

```
public class Wave {
    private int N;             // number of samples per channel
    private double[] left;     // samples for left channel
    private double[] right;    // samples for right channel
}
```

- The design of your program should look like Wave.java, except that you will need to fill in all of the constructors and methods.
- Your constructors for `Wave` will need to allocate and initialize two arrays of `double` using `new`. Observe that you have to do this in the constructor (and not when you declare the instance variables) since otherwise you wouldn't know how big to make the array.
- This file contains the doubles for the first 201 samples (approximately the first two sin waves in an A).

- Your `plus()` method should return a new `Wave` object so you will need to call the `Wave` constructor by using the keyword `new`. The method should not modify either of its two inputs in any way.

- For the echo filter, copy `MP3Player.java` to `EchoFilter.java`, and replace the body of `main()` to enable the echo effect. You will need to save the last 10 sound waves, so use an *array of waves* of size 10.

- The program `MP3Viewer.java` should be almost identical to `MP3Player.java`, except that you will call `draw()` instead of `play()`, and use `StdDraw.show()` to control the animation. Now make a `draw()` method for your Wave class. Invoking the method `draw()` should plot out both channels of that wave. To implement `draw()` use `StdDraw.point()` for each of the samples. Alternatively, you could use `StdDraw.line()`, but the animation may be substantially slower.

  It's convenient to rescale the coordinate system using `StdDraw.setXscale()` and `StdDraw.setYscale()`. Then, when you write `draw()`, it will be easy to scale `x` to match the sample index of the channel, and `y` to match the sample data (offset to be in the upper half for the left channel and lower half for the right channel). Feel free to use color when plotting.

Enrichment

Music instruments. A music instrument sounds more vibrant than the pure sine waves that we synthesize in Part 1. The reason for this is that music instruments generate not only the fundamental frequency f (as we did in Part 1), but also the higher *harmonics* 2f, 3f, 4f and so on. In addition, the note rises in intensity very quickly (but not instantaneously) and then gradually fades away. A simple way to model this is to multiply the amplitude at time t by e^-kt for some constant k.

felten.mp3. The song All My Trials was performed and recorded by Princeton professor Perry Cook after another Princeton professor Ed Felten was threatened with a lawsuit by the RIAA (Recording Industry Association of America) for presenting an academic paper that explained the inherent flaws in several digital watermarking schemes. Lifted by the spirits of Perry Cook's song, and with support from the EFF, Ed Felten was able to defeat the mighty RIAA.

MP3 file format. The MP3 file format stores digital audio in a compressed format. It compresses digital audio by eliminating (i) sounds that the human ear cannot hear, (ii) sounds that go unheard because there is another sound that masks it, and (iii) sounds that are not heard well at all. The net effect is a file that is 10-12 times smaller than the original, without sacrificing too much in fidelity. The mp3 file consists of *frames*. Each frame, when decoded, contains an array of 2,304 signed 16-bit integers that represent instantaneous amplitude. mp3 files interleave the left and right stereo channels: the first element of the array is the amplitude of the first sampling of the left stereo channel, and the second element is the amplitude of the first sampling of the right stereo channel, and so on.

Ripple echo effect. To get a ripple echo effect (where the sound echoes forever, but decays over time), take the average of the current `Wave` and the `Wave` you *played* 10 time steps ago (instead of the one you *read* in 10 steps ago).

Wave interference. Here's a fun and easy suggestion from a former student: set the right channel to be the opposite (negative) of the left channel; point your computer speakers at each other; and move them toward and away from each other. You should hear the effects of the waves cancelling each other out.

Ssllooww ddoowwnn ppllaayybbaacckk. Anothe fun thing to try is to play the wave in slow motion. Specifically create a `Wave` method `expand()` that returns a new `Wave` that is twice as long, and has each sample of the invoking `Wave` repeated twice, consecutively.

Frequencies of musical notes. Middle C is 240Hz. Each octave has a frequency that is twice the previous one. In Western culture, each octave is divided into twelve steps. Each note has a frequency that is the twelfth root of two times the previous one. This logarithmic scale correlates with human hearing.

Music theory. For a massive overload in information regarding Music Theory, check out http://www.musictheory.net.

More data files. Here are some famous speeches and soundbytes.

Java sound. Java Sound Resources contains code samples ranging from ripping CDs to MIDI synthesis.

COS 126 Home Page
wayne@cs.princeton.edu