Quick Exercise

Write a Sound method named `amplify( int change )` that increases the volume of the sound by `change` units.

Recall the availability of these methods of the Sound class:

```
getSamples()
getLength()
getSamplingRate()

getSampleValueAt( int slot )
setSampleValueAt( int slot, int newValue )
```

Alternative 1

How about this one?

```
public void amplify( int increase )
{
  for (int i = 0; i < this.getLength(); i++ )
  {
    int value = this.getSampleValueAt( i );
    this.setSampleValueAt( i, (int) (value + increase) );
  }
}
```

... do we notice a change?

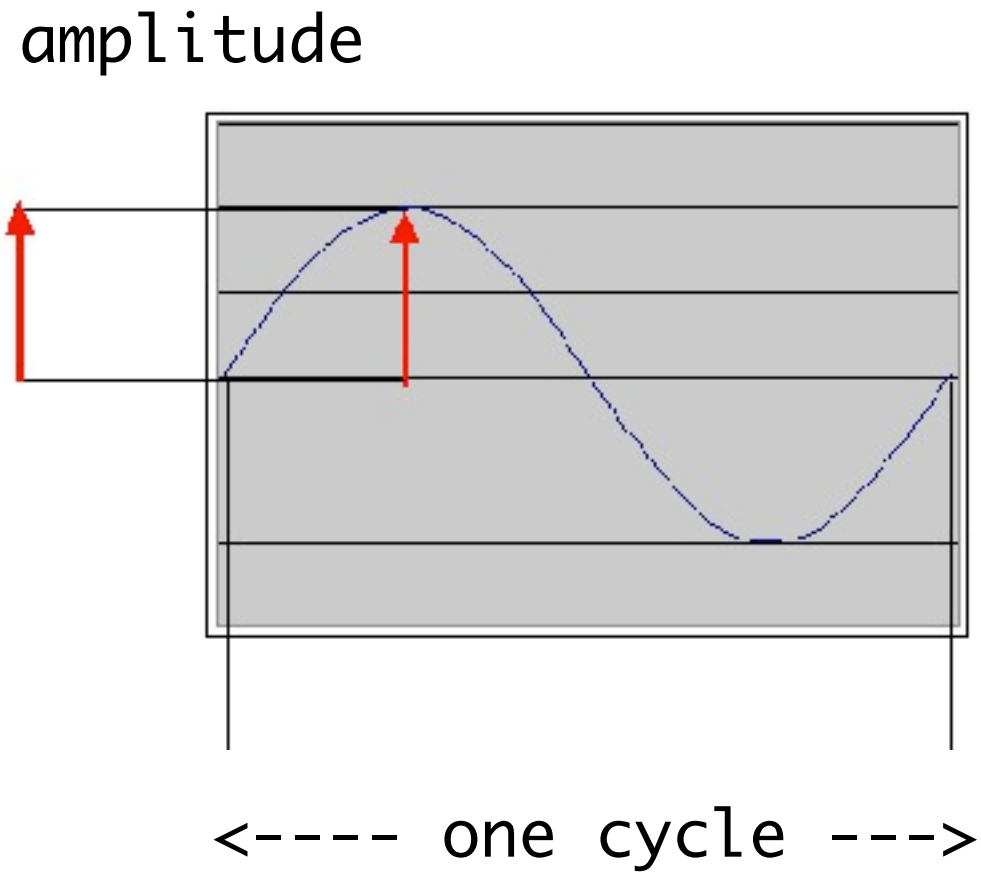
Alternative 2

How about *this* one?

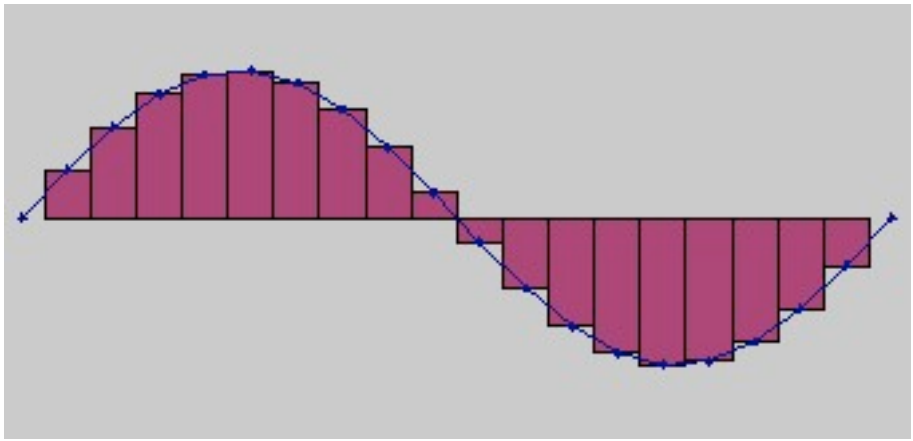
```
public void amplifyNotShift( int increase )
{
  for (int i = 0; i < this.getLength(); i++ )
  {
    int value = this.getSampleValueAt( i );
    if (value > 0 )
      this.setSampleValueAt( i, value + increase );
    else if (value < 0 )
      this.setSampleValueAt( i, value - increase );
  }
}
```

... what happens now?

Recall: The Physics of Sound



Digitizing Sound



We can estimate the area under a curve
using a sampling of rectangles.

To encode a sound,
we record the amplitude at a point in time —
the height of an implicit rectangle.

Computer Science Idea: File versus Object

A ***file*** is a unit stored on a drive somewhere.

We *open*, *read*, and *write* files.

An ***object*** is a unit associated with a program, in memory.

We can create an object by reading values from a file.
Changing the object does *not* change the file.

Quick Exercise

Write a Sound method named `invert()` that replaces all positive values with equivalent negative values, and vice versa.

Recall the availability of these methods of the Sound class:

```
getLength()

getSampleValueAt( int slot )
setSampleValueAt( int slot, int newValue )
```

Normalizing a Sound

The Idea

Multiply every sample by the same ratio
so that the largest value (positive or negative)
is the maximum possible value.

The result is to "stretch" the curve
upward and downward toward the maxima.

Normalizing a Sound

The Algorithm

1. Find the largest sample value in the sound, whether positive or negative.

2. Compute the normalizing factor: 32767/largest value.

3. Multiply every sample value by the factor.

How to Find the Largest Value?

How to Find the Largest Value?

Look at every value in turn.

If it is the largest we've seen so far,
remember it.

When you're done,
the largest we've seen so far
is the largest, period.

How to Find the Largest Value?

1. *Look at every value in turn.*
2. *If it is the largest we've seen so far, remember it.*
3. *When done, the largest we've seen so far is the largest.*

Before we start,
what is the largest value we've seen so far?

Initializing "Find Maximum"

1. Initialize 'largest so far' to the first value.
   Begin looking with the second value.

OR

2. Initialize 'largest so far' to the smallest possible value.
   Begin looking with the first value.

... sometimes, one of these works better than the other.