

```
1 class ArrayExamples {
2     public static void main(String[] args) {
3         int[] nums = {42, 56, 77};
4         int numsAt1 = nums[1];
5         nums[1] = 100;
6         int numsAt1After = nums[1];
7         System.out.println(numsAt1 + "\t" + numsAt1After);
8
9         int[] maybeCopy = nums;
10        maybeCopy[2] = 999;
11        System.out.println("Did nums[2] change?" + nums[2]);
12        System.out.println(nums);
13        System.out.println(maybeCopy);
14    }
15 }
```

ArrayExamples.java

at terminal

```
$ javac ArrayExamples.java
$ java ArrayExamples
```

Model 1

type	name	value
int	numsAt1	56
int	numsAt1After	100
int[]		
int[]		

Model 2

type	name	value
int[]	nums	
int[]	maybeCopy	

reference	value

<array>[<index>] = <value>

Array update or Array assignment:
Updates the array on the heap **referenced by** the <array> expression to have the given <value> at <index>.

<type>[] <name> = { <e1>, <e2>, ... };

Creates an **array** and stores it in the variable <name>.
All elements e1, e2, must have the given type.

<array>[<index>]

Array lookup:
Arrays can be indexed as in Python. <index> should evaluate to an int, and <array> to an array value. Indices start at 0.

int[]	CSE8ALib.readSound(String path) Takes a path to a file expected to be in .wav format and produces an int array representing the sound recorded in that file.
boolean	CSE8ALib.play(int[] sound) Takes an int array representing a sound and plays it (using the computer's speakers / headphones), returns true if the operation was performed successfully and false otherwise.
boolean	CSE8ALib.explore(int[] sound) Takes an int array representing a sound and opens a window that displays the sound waveform along with sampled values, returns true if the operation was performed successfully and false otherwise.

<pre>1 public class Sounds { 2 3 public static void main(String[] args){ 4 int[] sound = CSE8ALib.readSound("sounds/UpbeatFunk.wav"); 5 CSE8ALib.play(sound); 6 7 for(int i = 0; i < 1000; i += 1) { 8 System.out.print(sound[i] + " "); 9 } 10 } 11 12 }</pre>	<div>Sounds.java</div> <pre>\$ javac -cp lib/CSE8ALib.jar Sounds.java \$ java -cp lib/CSE8ALib.jar:. Sounds -15 0 24 0 48 0 -2 0 -8 0 -944 0 -3125 0 -2288 0 1988 0 5940 0 6448 0 3671 0 2240 0 3074 0 4576 0 5311 0 5048 0 1577 0 -100 0 -2206 0 639 0 4245 0 2748 0 -1465 0 - 5451 0 -6461 0 -5838 0 -4646 0 -4590 0 - 5454 0 -6426 0 -6092 0 -4555 0 -2967 0 - 2724 0 -2955 0 -3501 0 -3235 0 -2161 0 - 1275 0 -989 0 -1195 0 -1228 0 -534 0 234 0 709 0 470 0 650 0 679 0 1544 0 2239 0 2356 0 1950 0 1887 0 2173 0 2882 0 2805 0 2548 0 1943 0 2911 0 3012 0 3510 0 3212 0 4233 0 5429 0 650 0 4218 0 7176 0 3145 0 387 0 2572 0 10316 0 13406 0 11622 0 2154 0 -1661 0 -2069 0 -2513 0 842 0 -3045 0 - 7782 0 -5669 0 -8454 0 61 0 7493 0 -5511 0 -15136 0 -10269 0 -1196 0 8654 0 14108 0 13306 0 10621 0 9460 0 10155 0 5706 0 ...</pre>
---	--

	reduceVolume () Takes an array of integers representing a sound and returns a new array of integers representing the same sound, just quieter by a factor of 10.
--	---

```
int[] reduceVolume (int[] sound) {
    int size =

    int[] soundQuiet = new int[size];

    for(int i = 0; i < size; i += 1) {

        soundQuiet[i] =

    }
    return soundQuiet;
}
```