

CSE 8A Programming Assignment 5

Name should be formatted as (last, first)

If you are working solo you may leave the right column blank.

Name: Darren Yeung

Name:

PID: A15943292

PID:

Email: dyeung@ucsd.edu

Email:

Part 1 Testing Code:

(had three separate test methods)

// Copy and paste the code from only your test method (including comments!) here

// Make sure to set the font to **Courier New**

// **IMPORTANT:** Make sure your code is properly formatted and commented.

// Code that does not have correct indentation and comments will lose marks.

```
public static void test(){
    // test crop function
    int[] inputCropped = {3, 10, 5, 20, 9, 6};
    int[] croppedAns = {5, 20, 9};
    int[] cropped = crop( inputCropped, 2, 5 );
    System.out.print("Should be: ");
    printArray(croppedAns);
    System.out.print("Got: ");
    printArray(cropped);

    // test concatSounds
    int[] inputSoundConcat1 = {3, 10, 5};
    int[] inputSoundConcat2 = {7, -50, 13};
    int[] concatAns = {3,10,5,7,-50,13};
    int[] concatenated = concatSounds( inputSoundConcat1, inputSoundConcat2
);

    System.out.print("Should be: ");
    printArray(concatAns);
    System.out.print("Got: ");
    printArray(concatenated);

    // test sineSound
    int[] sinAns = {0, 43, -82, 115, -138, 149, -147, 133, -108, 73};
    int[] sineWave = sineSound(10, 10000, 150);
    System.out.print("Should be: ");
    printArray(sinAns);
```

```

        System.out.print("Got: ");
        printArray(sineWave);

        // test addSounds
        int[] inputSoundAdd1 = {3, 10, 5};
        int[] inputSoundAdd2 = {7, -50, 13};
        int[] addedAns = {10, -40, 18};
        int[] added = addSounds( inputSoundAdd1, inputSoundAdd2 );
        System.out.print("Should be: ");
        printArray(addedAns);
        System.out.print("Got: ");
        printArray(added);

    }

    public static void test2(){

        int[] inputCropped = {3, 10, 5, 20, 9, 6};
        int[] croppedAns = {20, 9};
        // making sure that it can stop at two numbers
        int[] cropped = crop( inputCropped, 3, 5 );
        System.out.print("Should be: ");
        printArray(croppedAns);
        System.out.print("Got: ");
        printArray(cropped);

        // test concatSounds
        int[] inputSoundConcat1 = {3, 10, 5,8};
        int[] inputSoundConcat2 = {7, -50, 13,8};
        //testing if code can handle array size four
        int[] concatAns = {3,10,5,8,7,-50,13,8};
        int[] concatenated = concatSounds( inputSoundConcat1, inputSoundConcat2
);

        System.out.print("Should be: ");
        printArray(concatAns);
        System.out.print("Got: ");
        printArray(concatenated);

        // test sineSound
        int[] sinAns = {0, 431, -872, 1152, -1380, 1490, -1475, 1334, -1081,
736};

        int[] sineWave = sineSound(10, 10000, 1500);
        //testing if amplitude can be greater than three digits
        System.out.print("Should be: ");
        printArray(sinAns);
        System.out.print("Got: ");
        printArray(sineWave);

        // test addSounds

```

```

int[] inputSoundAdd1 = {3, 10, 5};
int[] inputSoundAdd2 = {7, -500, 13};
int[] addedAns = {10, -450, 18};
//testing if method can handle negative values;
int[] added = addSounds( inputSoundAdd1, inputSoundAdd2 );
System.out.print("Should be: ");
printArray(addedAns);
System.out.print("Got: ");
printArray(added);

```

```

}

```

```

public static void test3(){

```

```

    int[] inputCropped = {3, 10, 5, 200, 9, 6};
    int[] croppedAns = {200, 9};
    // making sure method can handle three digit numbers
    int[] cropped = crop( inputCropped, 3, 5 );
    System.out.print("Should be: ");
    printArray(croppedAns);
    System.out.print("Got: ");
    printArray(cropped);

```

```

    // test concatSounds

```

```

    int[] inputSoundConcat1 = {3, 10, 5,8,10};
    int[] inputSoundConcat2 = {7, -50, 13,8,7};
    //making sure method can handle repeated numbers
    int[] concatAns = {3,10,5,8,10,7,-50,13,8,7};
    int[] concatenated = concatSounds( inputSoundConcat1, inputSoundConcat2

```

```

);

```

```

    System.out.print("Should be: ");
    printArray(concatAns);
    System.out.print("Got: ");
    printArray(concatenated);

```

```

    // test sineSound

```

```

    int[] sinAns = {0, -946, 1468, -1332, 600, 401, -1222, 1496, -1100, 210};
    int[] sineWave = sineSound(10, 13420, 1500);
    //testing to see if frequency can have digits other than zero after the

```

```

one

```

```

    System.out.print("Should be: ");
    printArray(sinAns);
    System.out.print("Got: ");
    printArray(sineWave);

```

```

    // test addSounds

```

```

    int[] inputSoundAdd1 = {3, 1000, 5};
    int[] inputSoundAdd2 = {7, -500, 13};
    int[] addedAns = {10, 500, 18};
    //testing the method to see if it can handle four digit numbeters;

```

```
        int[] added = addSounds( inputSoundAdd1, inputSoundAdd2 );
        System.out.print("Should be: ");
        printArray(addedAns);
        System.out.print("Got: ");
        printArray(added);
    }
```

Part 1 Test Results:

2.1 Show the results of running your test method here.

darrens-mbp:PA5 Starter rands\$ javac -cp lib/*:. Sounds.java

darrens-mbp:PA5 Starter rands\$ java -cp lib/*:. Sounds

Should be: {5, 20, 9 }

Got: {5, 20, 9 }

Should be: {3, 10, 5, 7, -50, 13 }

Got: {3, 10, 5, 7, -50, 13 }

Should be: {0, 43, -82, 115, -138, 149, -147, 133, -108, 73 }

Got: {0, 43, -82, 115, -138, 149, -147, 133, -108, 73 }

Should be: {10, -40, 18 }

Got: {10, -40, 18 }

Test 2:

Should be: {20, 9 }

Got: {20, 9 }

Should be: {3, 10, 5, 8, 7, -50, 13, 8 }

Got: {3, 10, 5, 8, 7, -50, 13, 8 }

Should be: {0, 431, -872, 1152, -1380, 1490, -1475, 1334, -1081, 736 }

Got: {0, 431, -827, 1152, -1380, 1490, -1475, 1334, -1081, 736 }

Should be: {10, -450, 18 }

Got: {10, -490, 18 }

Test 3:

Should be: {200, 9 }

Got: {200, 9 }

Should be: {3, 10, 5, 8, 10, 7, -50, 13, 8, 7 }

Got: {3, 10, 5, 8, 10, 7, -50, 13, 8, 7 }

Should be: {0, -946, 1468, -1332, 600, 401, -1222, 1496, -1100, 210 }

Got: {0, -946, 1468, -1332, 600, 401, -1222, 1496, -1100, 210 }

Should be: {10, 500, 18 }

Got: {10, 500, 18 }

2.2 Briefly explain why you chose each test and how you know your code is working correctly (or not).

Test1:

First method: had to make sure function can handle numbers of two digits

Second method: had to make sure function can handle numbers of two digits and also negative values

Third method: had to make sure method can handle negative values and also frequency of over four digits

Fourth Method: had to make sure method can sum up double digits and negative values.

Test2:

First method: making sure the crop method can stop at two number

Second method: testing if code can handle array size four

Third method: testing if amplitude can be greater than three digits

Fourth method: testing if method can handle negative values;

Test3:

First method: making sure method can handle three digit numbers

Second method: making sure method can handle repeated numbers

Third method: testing to see if frequency can have digits other than zero after the one

Fourth method testing the method to see if it can handle four digit numbers

Part 2: Sound Exploration

```
// Copy and paste the code from your main method (including comments!) here
```

```
// Make sure to set the font to Courier New
```

```
// IMPORTANT: Make sure your code is properly formatted and commented.
```

```
// Code that does not have correct indentation and comments will lose marks.
```

```
int[] firstsound = sineSound(30000, 10000, 10);
```

```
int[] secondsound = sineSound(20000, 25400, 10);
```

```
int[] thirdsound = concatSounds(firstsound, secondsound);
```

```
changeVolume(thirdsound, 10);

//boolean truefalse = lib.play( firstsound);
//System.out.println(truefalse);

lib.play(thirdsound);
lib.explore(thirdsound);


test();

System.out.println("");
System.out.println("");

System.out.println("Test 2: ");

test2();

System.out.println("");
System.out.println("");

System.out.println("Test 3: ");

test3();
```

Part 2 Description:

Describe the sound you produced and how you produced it. What did you like about this process? What challenges did you run into during this process and how did you resolve them?

Sound started at high frequency and then went to lower frequency. It stayed at the high frequency for around 1 second and then stayed at the lower frequency for like half a second. I produced it by using the sineSound method. I liked the creativity in this assignment. Everything else I did not like. Some challenges I ran into was not knowing how to use specific functions provided in the library. I resolved them by reading the documentation more carefully.