

CS 200 Sections 02 & 04 Spring 2013

Week #14: Top 3 Lessons Learned

Before you write anything down, separate the easy problems from the hard problems. This will give you more time to solve the hard ones later.

In one of the example for code tracing with array. we have `int [] bar = { 6, 4, 7, 1 };`

Before we start the output, we first create a memory box for the array.

We then label them from underneath starting from 0 to 3.

Cross off any number that has been changed and also make sure to keep track of the exact output.

Always remember that in the for loop M.O.E happens just before the last curly brace of the for loop, and the initial value is set only when entering the loop from the top.

As you trace through the code, make sure to relook the tracing one more time before time run out.

Another thing I learn was that it is best to look back to old problems/quizzes and use them as a review for the final. It is best to work out the problem and change the question of the problem.

Try to go back and relearn what "\n" mean and also the different between a+b and (a+b) within an output statement.

The last thing I learned is that It is best to prepare yourself ahead of time and try timing yourself when writing out the code. Recall all lessons from class or on the website page.

P. Khuu

When you are code-tracing it is important to take time to label the braces. This will help you to see where a loop ends and where it begins.

Second, on the day of the final, it is important that we print our name only on the first page of the final exam, and write our student identification number on each of the remaining pages of the exam.

Third, when it comes to multi-dimensional arrays, we are limited by the memory of a computer.

J. Gomez

The lessons are:

1. Arrays with 2 indices are called 2 dimensional arrays. It is basically a table of rows and columns.

2. An example for declaring and initializing a 2 dimensional array: `int [] [] table = new int [10] [6];` The 10 represents rows and the 6 represents columns.

3. Arrays can have many dimensions, a three dimensional array would be `int [] [] [] table = new int [10] [6] [3];` a four dimensional array would be written as `int [] [] [] [] table = new int [10] [6] [3] [4];`

Multidimensional arrays are limited by the amount of RAM in a computer.

D. Starostka

1. We do not stop at 2 or 3 dimensional-arrays, we could have indefinite dimensional-arrays, but in real world we do use some of them.

2. **The size of the array must be an integer.**

3. If we do not know exactly how many values do we need, just we declare an array large enough to hold the largest possible number of items. Or if the user is going to define the size of the array we must do the process in this order:

```
int [ ] userDef;
```

```
int size; //size of array must be of integer type and must be a positive integer
```

```
System.out.print("Please enter an integer for size of array you need: ");
```

```
size = kbd.nextInt( ); //get size of array from user
```

```
userDef = new int[size];
```

D. Mirdadi