

CSC 205 Lab 4 : Multi-Dimensional Arrays & ArrayList ADT

This lab should be completed in class on Monday, September 19th.

Goals

After completing this lab, you should be able to:

- Write class methods for processing multi-dimensional arrays.
- Understand the advantages and disadvantages of vectors as compared to arrays.
- Use all of the common methods of the ArrayList ADT appropriately.

Lab Setup

Change into your Labs directory, and let's create and change into a Lab3 directory.

Now, let's copy over some files by typing :

```
cp /pub/digh/CSC205/Lab4/* .
```

Multi-dimensional Arrays

1. Compile and run program Table. Notice how we have a 3 by 3 square matrix whose elements represent the sum of their indices within the matrix.

	[0]	[1]	[2]
[0]	0	1	2
[1]	1	2	3
[2]	2	3	4

Now, let's take the elements that are currently contained in our matrix and multiply each one by a scale of `factor` where `factor` is an integer passed in as a parameter. Add a void class method named `scale` whose header is as follows.

```
private static void scale(int[][] table, int factor)
```

Include a call to your method in main with a `factor` of 5, and re-print your matrix. You should now have this matrix printed to the screen.

	[0]	[1]	[2]
[0]	0	5	10
[1]	5	10	15
[2]	10	15	20

2. Declare another 3 by 3 square matrix in main named `myTable` and initialize its components to all 1 using aggregate assignment. For example,

```
int[][] myTable = {{1, 1, 1},  
                  {1, 1, 1},  
                  {1, 1, 1}};
```

ArrayLists

1. Trace through the LabArrayList program below and determine what will be printed to the screen. Draw your vector at each stage in your notebook and write down your final answers.

```
import java.util.*;

public class LabArrayList
{
    public static void main(String[] args)
    {
        ArrayList<String> a = new ArrayList<String>(3);
        System.out.println(a.size());

        a.add("jazz");
        a.add("rock");
        a.add("top40");
        a.add(2, "metal");
        a.set(1, "classical");
        a.remove("jazz");
        a.add("country");
        a.add(a.get(a.size()-1));

        System.out.println(a.size());
        for (int i = 0; i < a.size(); i++)
            System.out.print(a.get(i) + " ");
        System.out.println();
        System.out.println(a.indexOf("metal"));
        System.out.println(a.contains("jazz"));

        String str = (String) a.get(0);
        Object obj = a.get(a.size()-1);
        System.out.println(str + " " + obj);

        System.out.println(a.size());
    }
}
```

2. When you're done, compile and test your program to check your answers. Now, answer the following questions on vectors.

a) What exactly is the difference between the size of an ArrayList and the capacity of the ArrayList? *Capacity = ability to store*

Size = can actually store

b) How many elements can an ArrayList store?

Infinite, capped at the data storage

c) Name two advantages of ArrayLists over arrays.

ArrayLists can increase/decrease capacity, and take up less memory

d) Name two advantages of arrays over ArrayLists.

Easier to remove elements in arrays over ArrayLists

Less memory is used if elements are used beforehand

3. Create a program named `TestArrayList`, and declare an `ArrayList` with an initial capacity of 10. Insert one copy of the string "Python", five copies of the string "Java" followed by four copies of the string "C++" so that your vector is now filled to capacity.

- a) Add a class method named `printArrayList` to print out all the elements stored within an `ArrayList` on separate lines. Test your method to be sure it works.

- b) Add a class method named `delete` to remove all copies of a particular string object passed in as a parameter. The parameter key should be of type `String`. That is,

```
private static void delete(ArrayList<String> myVector,  
    String key)
```

Comment out your first print call, and now include a call to `delete(myArrayList, "Java")` to test your `delete` method. Include a call to `printVector` following it to verify that only five string objects now remain.

- c) Add a value-returning class method named `count` that can be used to count the number of times that a particular string object passed in as a parameter occurs within an `ArrayList`. Include a call to `count` in main to test your method.