

# LAB 3 (ARRAYS, ARRAYLISTS, ITERATORS, AND 2048 HELPER)

CSC 172 (Data Structures and Algorithms)

Spring 2018

University of Rochester

**Due Date: Sunday, February 11 @ 11:59 pm**

## Introduction

The labs in CSC172 will follow a pair programming paradigm. Every student is encouraged (but not strictly required) to have a lab partner.

Every student must hand in their own work, but every student must list the name of their lab partner if any on all labs.

## Objectives

Explore Arrays and ArrayLists in more details. Learn to work with iterators. This lab has some elements common to Project 1 (Yes, that's intentional!). If you have a partner and you both end up using a portion of this code in your project, you must need to state that in the README file.

## Task 1

Implement a function `print2Darray` to print a formatted 4x4 two dimensional integer array.

When the array contains `{{10, 15, 30, 40},{15, 5, 8, 2}, {20, 2, 4, 2},{1, 4, 5, 0}}`, Your output should look like:

```
10  15  30  40
15   5   8   2
20   2   4   2
1    4   5   0
```

Now, Implement another function `print2DList` to print a formatted 2D list

```
ArrayList< ArrayList< Integer > > array;
```

## Task 2

Implement a method/function that produce running sum `runningSum2DArray` across rows (left to right or right to left) or column (top to bottom or bottom to top)

**Input to the method:** A 4x4 two dimensional int array and an integer (1,2,3 or 4 for left, right, up, down respectively).

**Output:** The modified array after producing the running sums according to the direction.

For example: If the input to the method is the same as the earlier array, and if the direction is 2 (right), the output array would be printed as:

```
10  25  55  95
15  20  28  30
20  22  26  28
1   5  10  10
```

Now, Implement another function `runningSum2DArrayList` that performs the same functionality, but takes `ArrayList< ArrayList< Integer > > array` as input.

## Task 3 (Iterate over an ArrayList)

In Java, you can iterate an ArrayList in different ways. Write the following methods to print Integers in an ArrayList iterating in different ways:

1. 

```
// Using basic while / for loop
void printArrayListBasicLoop(ArrayList<Integer> al);
```
2. 

```
// Using enhanced for loop (:)
void printArrayListEnhancedLoop(ArrayList<Integer> al);
```
3. 

```
// Using basic for loop with iterator
void printArrayListForLoopListIterator(ArrayList<Integer> al);
```
4. 

```
// Using basic while loop with iterator
void printArrayListWhileLoopListIterator(ArrayList<Integer> al);
```
5. 

```
// Using Java 8 forEach with lambda (use ArrayList.forEach method)
void printArrayListLambda(ArrayList<Integer> al);
```

Note: You can find related code from: <https://memorynotfound.com/iterate-arraylist-java/>. The purpose of this exercise is to make you familiar with iterator and understanding how to use them. In Lab 5, you have to implement iterators on your own. We will discuss iterator in more details before that in class.

## Submission

Hand in the source code from this lab at the appropriate location on the Blackboard system at `learn.rochester.edu`. You should hand in a single zip (compressed archive) `Lab3.zip` containing your source code and README files, as described below. Your source code should include three Java files:

`Lab3Task1.java`, `Lab3Task2.java` and `Lab3Task3.java`.

1. A plain text file named README that includes your contact information, your partner's name, a brief explanation of the lab (a one paragraph synopsis. Include information identifying what class and lab number your files represent.), and one sentence explaining the contents of any other files you hand in.
2. Three Java source code file representing the work accomplished. All source code files should contain author and partner identification in the comments at the top of the file. For Task 2, you can reuse methods from Task 1.

## Grading (100 pts)

Task 1: 30 pts

Task 2: 40 pts

Task 3: 30 pts

## Notes:

All labs are open book. You can get code snippets from the internet if you want (make sure you cite those properly). But that is not the purpose. We want you to sit together, think about an algorithm, and then implement it together with your partner.