

Lab: Arrays

Tutorial Lab 1: Using an Array

An array is a data structure that groups data together into a collection of **elements**. Each element has its own **index** or position within the array. These elements can be initialized, accessed, modified, and printed. Copy the code below into the text editor on the left. Then click on the TRY IT button to see the resulting output and ++Code Visualizer++ link underneath to see how the program runs line by line.

```
String[] classes = {"Math", "English", "Science", "Social  
Studies", "Spanish"};  
  
classes[4] = "French";  
  
for (int i = 0; i < classes.length; i++) {  
    System.out.println(classes[i]);  
}
```

[Code Visualizer](#)

Program Summary

1. An array called `classes` is created and initialized with elements (i.e. "Math", "English", etc.).
2. At index 4, "Spanish" is modified and replaced with "French". Keep in mind that array indices start at 0, not 1.
3. A for loop iterates through the elements in the array. It starts at index 0, ends right before index 5, and increments by 1 after each iteration.
4. Each element from index 0 through index 4 gets printed with a newline.

Lab: ArrayLists

Tutorial Lab 2: Using an ArrayList

An ArrayList is another data structure that has many of the same functionalities as an array. Though ArrayLists cannot be initialized at the same time as they are created, they are flexible in the methods that they are able to use. These methods include adding and removing elements within the ArrayList, meaning ArrayLists can **dynamically** change their size, something arrays cannot do. Copy the code below into the text editor on the left. Then click on the TRY IT button to see the resulting output.

```
ArrayList<String> veggies = new ArrayList<String>();
veggies.add("carrot");
veggies.add("tomato");
veggies.add("celery");
veggies.add("spinach");

veggies.remove(1);
veggies.set(1, "potato");

for (String s : veggies) {
    System.out.println(s);
}
```

Program Summary

1. An ArrayList called veggies is created.
2. carrot, tomato, celery, and spinach are added to the ArrayList as elements.
3. The element at index 1 (tomato) is removed.
4. The element potato replaces the element at index 1, which is currently celery since tomato was deleted previously.
5. An enhanced for loop is used which creates a String iterating variable s to take on the value of each element.
6. Each element s is then printed with a newline.

Lab: 2D Arrays

Tutorial Lab 3: Using a 2D Array

A 2D array is an array inside another array. This data structure can be visually compared to a table where there are rows and columns and each element exists inside each “cell.” To access or modify elements, both the row index and column index are needed. Like arrays, 2D arrays are **static** so elements cannot be added or removed after initialization.

```
int[][] coordinates = { {-4, 3},
                        {2, 1},
                        {0, -8},
                        {-11, 9},
                        {-5, -7} };

for (int i = 0; i < coordinates.length; i++) {
    for (int j = 0; j < coordinates[0].length; j++) {
        if (j == 1) {
            System.out.println(coordinates[i][j]);
        }
        else {
            System.out.print(coordinates[i][j] + " ");
        }
    }
}
```

[Code Visualizer](#)

Lab Challenge: 2D Chessboard

2D Chessboard

You are trying to create a chessboard representation using the alphabetical letters 0 and x. The 0 represents the lighter spaces while the x represents the darker spaces.

```
0X0X0X0X
X0X0X0X0
0X0X0X0X
X0X0X0X0
0X0X0X0X
X0X0X0X0
0X0X0X0X
X0X0X0X0
```

So far you have the following code within the text editor to your left:

```
public class LabChallenge {
    public static void main(String args[]) {

        String[][] chessboard = new String[8][8];

        //add code below this line

        //add code above this line

        for (int row = 0; row < chessboard.length; row++) {
            for (int col = 0; col < chessboard[0].length; col++) {
                if (col == chessboard[0].length - 1) {
                    System.out.println(chessboard[row][col]);
                }
                else {
                    System.out.print(chessboard[row][col]);
                }
            }
        }
    }
}
```

Chessboard Challenge

challenge

Assignment:

For this challenge, you will use your knowledge of 2D arrays to produce the chessboard pattern:

```
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
OXOXOXOX
XOXOXOXO
```

Requirement:

Your program **cannot make any changes** to the existing code in the program. If you do, you will **not earn any credit** for this challenge. If you accidentally delete any existing code, you can copy the original code shown above back into your program.

Hint: It is probably much easier to use nested `for` loops in your code to populate the 2D array with either 0 or x than to go through each (row, column) index to modify the elements.