Week 2-Lab#1

Section: 103
Simge Tekin

About me

- From Ankara, Turkey
- stekin@umd.edu
- 2nd-year CS PhD student
- Area: Data-driven security
- Office hours: Monday 4-5pm (Zoom), Tuesday 1-5pm

Project-1 is out!

Deadline: **Tue, 11 Feb at 11:31 PM** (Late: **Wed, 12 Feb at 11:31 PM**)

You will implement a Zoo package, with classes Animal, Lion, Eagle, Snake, RattleSnake, Habitat, and Zoo.

Details: Course page > Projects > Project #1 (Zoo)

Project-1 is out!

Deadline: Tue, 11 Feb at 11:31 PM (Late: **Wed, 12 Feb at 11:31 PM**)

You will implement a Zoo package, with classes Animal, Lion, Eagle, Snake, RattleSnake, Habitat, and Zoo.

Download Zoo.zip from the project page. It includes:

- A package named **zoo**
- A package named **tests**

Types of tests

Public tests: They are provided to you, located under test package of project-1

Study the public tests to understand what exactly you are required to implement.

Release tests: their result become available after your submission passes all the public tests. You get 3 token per day to utilize the release tests.

Secret tests: You do not know what the tests are about and you will not be able to see their result. No secret tests for P1.

Project-1 is out!

- Start to submit early, so that you have a backup
- Highest scoring submission will be graded
  - Be aware that this only applies to implementation, documentation will be graded separately.
- Don’t forget to clearly comment the code, documentation is crucial for you and others to understand your code.
- For style guidelines, you can take a look at:
  - https://www.cs.umd.edu/~nelson/classes/resources/javastyleguide/

ArrayLists- Recap

ArrayList class provides a dynamic way to store collections of objects, offering flexibility and ease of use compared to traditional arrays.

Dynamic Size: Unlike arrays, ArrayLists can grow or shrink in size as needed, allowing for efficient memory management.

Object Reference Storage: ArrayLists store references to objects, meaning you can store elements of any data type (including primitives through wrappers).

Index-Based Access: Elements can be accessed using their index, similar to arrays.

Built-in Methods: ArrayLists offer a rich set of methods for adding, removing, searching, and manipulating elements.

ArrayLists- Recap

Don’t forget to `import java.util.ArrayList;`

Creating the list:

```
ArrayList<String> al = new ArrayList<>();
```

Adding elements:

- `al.add("CMSC");`
- `al.add(1, "132"); //` Adds element at a specific index

  `// IndexOutOfBoundsException` - if the index is out of range

Removing elements:

- `al.remove(1);` **// Using index**
- `al.remove("CMSC");` **// Using value al.remove(Integer.valueOf(1))**

ArrayLists- Recap

Setting elements:

- `al.set(0, "GFG");` **//update the value at index 0**

Getting elements:

- `al.get(0);`

ArrayLists- Recap

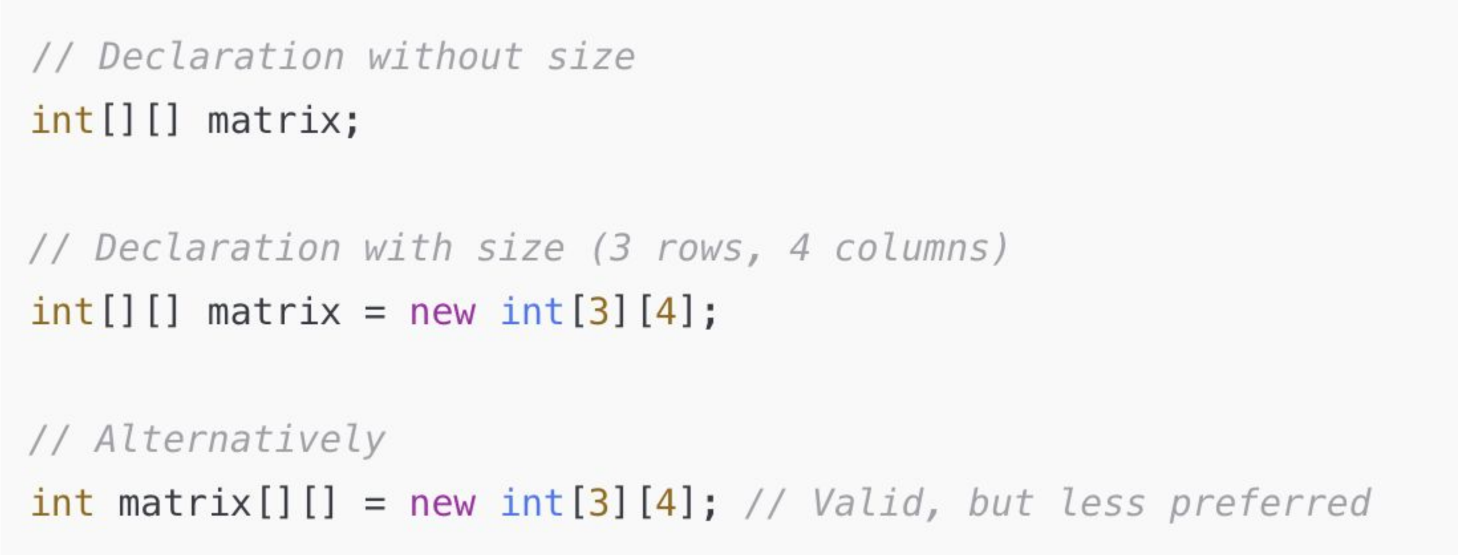
`clear()`: removes all the elements

`size()`: returns the number of elements in ArrayList

`clone()`: creates a shallow copy of the ArrayList; It only copies the references to the objects in the original list. If you modify the objects inside the cloned list, the changes will be reflected in the original list as well.

See the documentation for more: https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html

2D Arrays - Recap



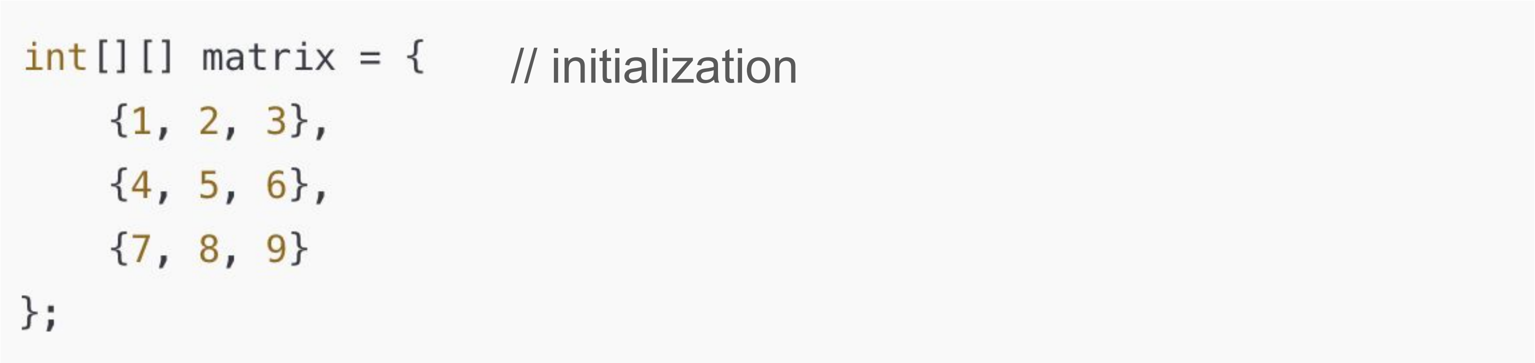
```
// Declaration without size
int[][] matrix;

// Declaration with size (3 rows, 4 columns)
int[][] matrix = new int[3][4];

// Alternatively
int matrix[][] = new int[3][4]; // Valid, but less preferred
```

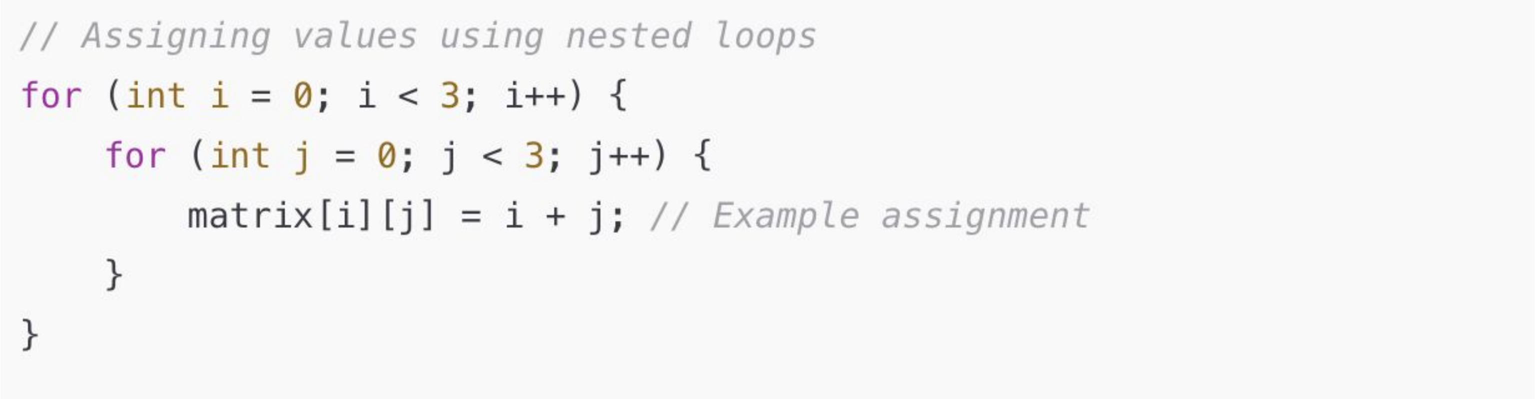
matrix is of type **int[ ][ ]**, matrix[0] is of type **int[]**, and matrix[0][0] is an **int**

2D Arrays - Recap



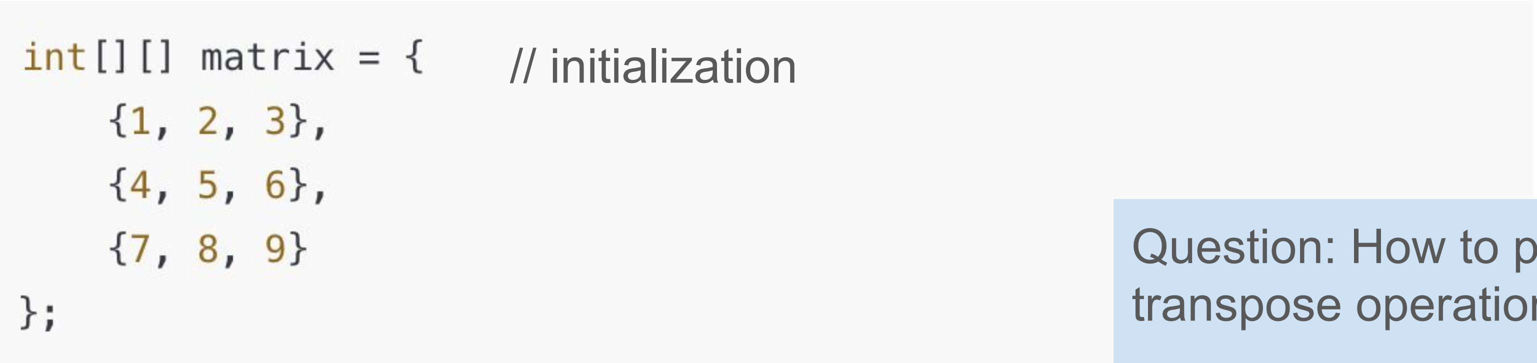
```
int[][] matrix = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};
```

// initialization



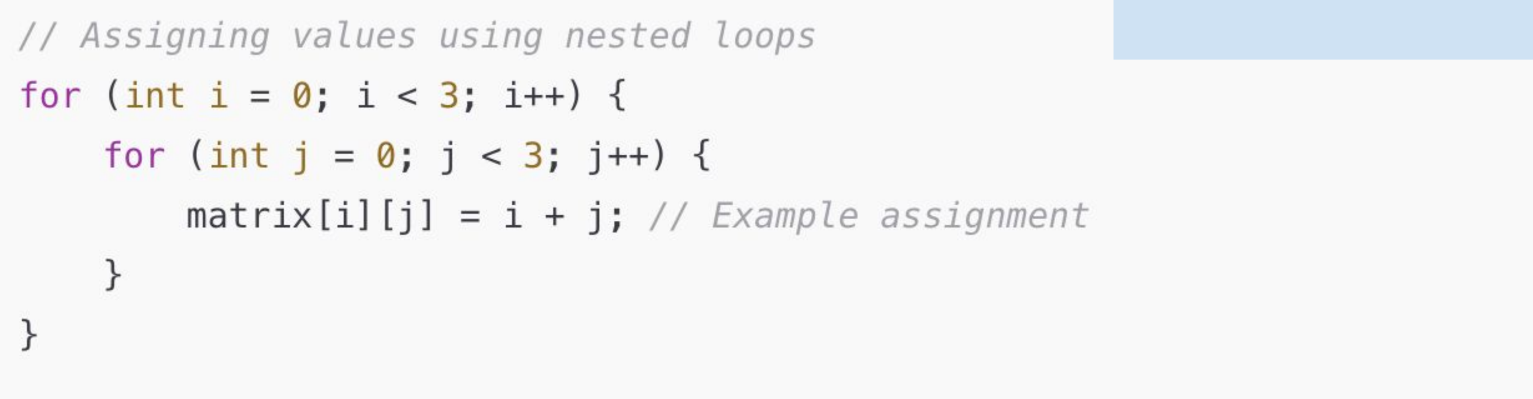
```
// Assigning values using nested loops
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        matrix[i][j] = i + j; // Example assignment
    }
}
```

2D Arrays - Recap



```java
int[][] matrix = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9}
};
```

// initialization



```java
// Assigning values using nested loops
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        matrix[i][j] = i + j; // Example assignment
    }
}
```

Question: How to perform transpose operation?