Yunu Jung (정윤우)
15655197127

AP Computer Science A

Lesson 3

Array

- A fixed-size collection of elements of the same type, stored in contiguous memory.
- Declared with brackets, e.g., int[] numbers = new int[5]; creates an array of 5 integers.
- Indexed from 0 to length - 1.

Array Initialization

- Can be initialized with specific values: int[] nums = {1, 2, 3, 4};.
- Default values: 0 for numbers, false for booleans, null for object references.

Array Length

- Accessed using the .length field (not a method).
- Example: numbers.length gives the number of elements in the array.

Array Traversal

- Access elements with a for loop:
- for (int i = 0; i names = new ArrayList();.

ArrayList Methods

- add(element) ➜ Inserts an element at the end.
- get(index) ➜ Retrieves the element at a given position.
- set(index, element) ➜ Replaces the element at a position.
- remove(index) ➜ Deletes the element at a position.
- size() ➜ Returns the number of elements in the list.

Generics in ArrayList

- ArrayLists must specify a type in angle brackets: ArrayList.
- Prevents mixing different data types in the same list.

2D Array

- An array of arrays (a grid-like structure).
- Declared as int matrix = new int3; ➔ 3 rows, 4 columns.

2D Array Initialization

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

2D Array Access

- Elements are accessed with two indices: matrixrow.
- First index = row, second index = column.

2D Array Traversal

```
for (int r = 0; r < matrix.length; r++) {
    for (int c = 0; c < matrix[r].length; c++) {
        System.out.print(matrixr + " ");
    }
}
```