AP Computer Science A Study Guide Unit 8

From Simple Studies: https://simplestudies.edublogs.org & @simplestudiesinc on Instagram

2D Array

2D Arrays

- 2D arrays are stored as arrays of arrays.
    - 2D arrays are created and indexed similar to 1D array objects.
- 2D arrays are declared and created with this syntax:
    - datatype[][] variableName = new datatype[numberRows][numberCols];
- 2D array objects that are not rectangular are not part of the course and AP Exam.
- For the purposes of the exam, when accessing the element at arr[first][second]:
    - The first index is used for rows, the second index is used for columns.
- The initializer list used to create and initialize a 2D array consists of initializer lists that represent 1D arrays.
    - For example, int[][] ticketInfo = { {45,20,45}, {45,20,45} };
- The square brackets [row][col] are used to access and modify an element in a 2D array.

2D Arrays (continued)

- "Row-major order"

Ordering 2D array elements where traversal occurs across each row
While “column-major order” traversal occurs down each column.

```
public class TwoDArrayInitGet

{

  public static void main(String[] args)

  {

    String[][] seating = { {"Sarah", "Maria"},

                           {"Cameron", "Ella"},

                           {"Emma", "Luke"} };

    String name = seatingInfo[0][0];

    System.out.println(name + " is at [0,0]");

  }

}
```

Traversing 2D Arrays

- 2D arrays use nested for loops or nested enhanced for each loop.
- Outer loop for a 2D array usually traverses the rows of an array.
    - The inner loop traverses through the columns.

- Nested iteration
    - Row-major order
    - Column-major order
- Standard algorithms for 1-dimensional Arrays can also be used for 2-dimensional Arrays.
- Each row must be accessed in a 2D Array for:
    - Sequential searches
    - Linear searches

The information below is from Runestone Academy and it is specified that this is tested on the AP exam!

- **2d Array** - An array that holds items in a two-dimensional grid. You can think of it as storing items in rows and columns (like a bingo card or battleship game). You can access an item (element) at a given row and column index.
- **2d Array Declaration** - To declare an array, specify the type of elements that will be stored in the array, then ([][]) to show that it is a 2d array of that type, then at least one space, and then a name for the array. Examples: int[][] seats; String[][] seatingChart;
- **2d Array Creation** - To create a 2d array, type the name and an equals sign then use the new keyword, followed by a space, then the type, and then [numRows][numCols]. Example: seatingChart = new String[5][4];. This will have 5 rows and 4 columns.
- **2d Array Index** - You can access and set values in a 2d array using the row and column index. The first element in an array called arr is at row 0 and column 0 arr[0][0].

Traversing 2D Arrays (continued)

- **2d Array Initialization** - You can also initialize (set) the values in the array when you first create it. In this case you don't need to specify the size of the array, it will be determined from the number of values that you specify.
  Example: String[][] seatingInfo = { {"Jamal", "Maria"}, {"Jake", "Suzy"}, {"Emma", "L uke"}}; This will create a 2d array with 3 rows and 2 columns.
- **2d Array Number of Rows** - The number of rows (or height) is the length of the outer array. For an array arr use arr.length to get the number of rows in the array.

- **2d Array Number of Columns -** The number of columns (or width) is the length of the inner array. For an array arr use arr[0].length to get the number of columns.
- **nested for loop -** A for loop inside of another for loop. These are used to loop through all the elements in a 2d array. One loop can work through the rows and the other the columns.
- **out of bounds error -** This happens when a loop goes beyond the last valid index in an array. Remember that the last valid row index is arr.length - 1. The last valid column index is arr[0].length - 1.