# UNIT 8

# 2D Array

**7.5–10%**
AP EXAM WEIGHTING

**~10–12**
CLASS PERIODS

**AP**

Remember to go to **AP Classroom** to assign students the online **Personal Progress Check** for this unit.

Whether assigned as homework or completed in class, the **Personal Progress Check** provides each student with immediate feedback related to this unit's topics and skills.

## Personal Progress Check 8

**Multiple-choice: ~10 questions**
**Free-response: 1 question**

- 2D Array

# 2D Array

## Developing Understanding

### BIG IDEA 1
*Variables* **VAR**

- Why might you want to use a 2D array to store the spaces on a game board or the pixels in a picture, rather than a 1D array or `ArrayList`?

### BIG IDEA 2
*Control* **CON**

- Why does the order in which elements are accessed in 2D array traversal matter in some situations?

In Unit 6, students learned how 1D arrays store large amounts of related data. These same concepts will be implemented with two-dimensional (2D) arrays in this unit. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplished this is in row-major order, but it is important to cover additional traversal patterns, such as back and forth or column-major.

## Building Computational Thinking Practices

**1.B** **2.B** **2.D** **3.E**

Students should be able to determine the result of program code that traverses and manipulates the elements in a 2D array. Traversals of 2D arrays typically require a set of nested loops. Often the extra dimension of a 2D array is difficult for students to envision. Providing students with practice analyzing and tracing traversals of a 2D array, as well as providing them partial program code to complete, helps students take this more abstract concept and make it concrete and replicable.

Because 2D arrays are traversed using nested loops, the number of times a code segment executes is multiplied. In a nested loop, the inner loop must complete all iterations before the outer loop can continue. It helps to provide students with sample code that will print the values in a 2D array. Teachers can an IDE that shows access to 2D arrays visually and keeps track of the execution count.

## Preparing for the AP Exam

The free-response portion of the exam always includes one question that requires students to write program code involving 2D arrays. Because 2D arrays are arrays where each element is an array, it is not uncommon for the question to require students to write solutions involving array or `ArrayList` objects as well.

While there is a specific nested structure to traverse elements in a 2D array in row-major order, this structure can be modified to traverse 2D arrays in other ways, such as column-major, by switching the nested iterative statements. Additional modifications can be made to traverse rows or columns in different ways, such as back and forth or up and down. However, when making these adjustments, students often neglect to adjust the bounds of the iterative statements appropriately. Students should practice traversing 2D arrays in these nonstandard ways, being sure to test the boundary conditions of the iterative statements, to be prepared for this type of free-response question.

# UNIT AT A GLANCE

| Enduring Understanding | Topic | Suggested Skills | Class Periods<br>~**10−12** CLASS PERIODS |
|---|---|---|---|
| VAR-2 | **8.1** 2D Arrays | **1.B** Determine code that would be used to complete code segments.<br><br>**1.C** Determine code that would be used to interact with completed program code.<br><br>**3.E** Write program code to create, traverse, and manipulate elements in 2D array objects. | |
| VAR-2, CON-2 | **8.2** Traversing 2D Arrays | **2.B** Determine the result or output based on statement execution order in a code segment without method calls (other than output).<br><br>**2.D** Determine the number of times a code segment will execute.<br><br>**3.E** Write program code to create, traverse, and manipulate elements in 2D array objects.<br><br>**4.A** Use test-cases to find errors or validate results. | |

**AP** Go to **AP Classroom** to assign the **Personal Progress Check** for Unit 8.
Review the results in class to identify and address any student misunderstandings.

# SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics in this unit. Please refer to the Instructional Approaches section beginning on p. 159 for more examples of activities and strategies.

| Activity | Topic | Sample Activity |
|---|---|---|
| 1 | 8.1 | **Using manipulatives**<br>Use different-sized egg cartons or ice cube trays with random compartments filled with small toys or candy. Create laminated cards with the code for the construction of, and access to, a 2D array, leaving blanks for the name and size dimensions. Have students fill in the missing code that would be used to represent the physical 2D array objects and access the randomly stored elements. |
| 2 | 8.2 | **Activating prior knowledge**<br>When first introducing 2D arrays and row-major traversal, ask students which part of the nested `for` loop structure loops over a 1D array. Based on what they know about the traversal of 1D array structures, ask them to calculate the number of times the inner loop executes. |
| 3 | 8.2 | **Sharing and responding**<br>As a class, create a set of test cases to be used with answers to a free-response question. Have students write their answers to the free-response question individually on paper. After exchanging solutions with another student, ask students to find errors or validate results of their peers' code by tracing the code with the developed test cases. Allow students an opportunity to provide feedback on the program code as well as the results of each test case. |

## ✎ Unit Planning Notes

*Use the space below to plan your approach to the unit. Consider how you want to pace your course and where you will incorporate writing and analyzing program code that represents 2D data sets.*

......................................................................................................................................................................

......................................................................................................................................................................

......................................................................................................................................................................

{ } After completing this unit, students will have covered all of the necessary content for the Steganography Lab. The proposed class periods for this unit include time to complete the provided lab activities.

# TOPIC 8.1
# 2D Arrays

## Required Course Content

### ENDURING UNDERSTANDING

**VAR-2**

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

### LEARNING OBJECTIVE

**VAR-2.F**

Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.

### ESSENTIAL KNOWLEDGE

**VAR-2.F.1**

2D arrays are stored as arrays of arrays. Therefore, the way 2D arrays are created and indexed is similar to 1D array objects.

> ☒ **EXCLUSION STATEMENT—**(EK VAR-2.F.1): 2D array objects that are not rectangular are outside the scope of the course and AP Exam.

**VAR-2.F.2**

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.

**VAR-2.F.3**

The initializer list used to create and initialize a 2D array consists of initializer lists that represent 1D arrays.

**VAR-2.F.4**

The square brackets `[row][col]` are used to access and modify an element in a 2D array.

**VAR-2.F.5**

"Row-major order" refers to an ordering of 2D array elements where traversal occurs across each row, while "column-major order" traversal occurs down each column.

## TOPIC 8.2
# Traversing 2D Arrays

## Required Course Content

### ENDURING UNDERSTANDING

**VAR-2**

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

### LEARNING OBJECTIVE

**VAR-2.G**

For 2D array objects:
- Traverse using nested `for` loops.
- Traverse using nested enhanced `for` loops.

### ESSENTIAL KNOWLEDGE

**VAR-2.G.1**

Nested iteration statements are used to traverse and access all elements in a 2D array. Since 2D arrays are stored as arrays of arrays, the way 2D arrays are traversed using `for` loops and enhanced `for` loops is similar to 1D array objects.

**VAR-2.G.2**

Nested iteration statements can be written to traverse the 2D array in "row-major order" or "column-major order."

**VAR-2.G.3**

The outer loop of a nested enhanced `for` loop used to traverse a 2D array traverses the rows. Therefore, the enhanced `for` loop variable must be the type of each row, which is a 1D array. The inner loop traverses a single row. Therefore, the inner enhanced `for` loop variable must be the same type as the elements stored in the 1D array.

### SUGGESTED SKILLS

**2.B**

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

**2.D**

Determine the number of times a code segment will execute.

**3.E**

Write program code to create, traverse, and manipulate elements in 2D array objects.

**4.A**

Use test-cases to find errors or validate results.

### AVAILABLE LABS
- Classroom Resources >
  - **AP Computer Science A: Picture Lab**
  - **AP Computer Science A: Steganography Lab**

### AVAILABLE RESOURCES
- **Runestone Academy: AP CSA—Java Review: 10.7—Looping through a 2D Array**
- **Practice-It!: BJP4 Chapter 7: Arrays—Exercises 7.19–7.19**
- The Exam >
  - **2017 AP Computer Science A Exam Free-Response Question #4 (Position)**
  - **2018 AP Computer Science A Exam Free-Response Question #4 (ArrayTester)**
  - Past AP Exam Questions on 2D Arrays on AP Question Bank

## ENDURING UNDERSTANDING

CON-2

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

### LEARNING OBJECTIVE

CON-2.N

For algorithms in the context of a particular specification that requires the use of 2D array traversals:

- Identify standard algorithms.
- Modify standard algorithms.
- Develop an algorithm.

### ESSENTIAL KNOWLEDGE

CON-2.N.1

When applying sequential/linear search algorithms to 2D arrays, each row must be accessed then sequential/linear search applied to each row of a 2D array.

CON-2.N.2

All standard 1D array algorithms can be applied to 2D array objects.