

# LATHROP ENGINEERING

Name: \_\_\_\_\_

## UNIT 5: ARRAYS














AP Computer Science A

Unit Due Date: **November 27, 2019**

Welcome to the fifth unit of *AP Computer Science*! In this unit, we'll get to continue developing the relationships between classes, objects, and methods while we also add a new tool to our program repertoire: *arrays*. Arrays – in a simple way – are lists of things, and will allow us to store multiple values in just one variable! In the end, the expectation is that you learn the following:

- How to organize classes, objects, and methods to make a program work well
- How to create arrays of different types
- How to traverse an array using a for loop
- How to create multidimensional arrays
- How to traverse multidimensional arrays with nested for loops

As we move through this unit, you are responsible for making adequate progress through the assignments, and for being done by the Unit Due Date (**November 27, 2019**). You are also responsible for completing each part before moving on to the next. Our unit is broken up into three main parts:

<b>Part 1: One-Dimensional Arrays</b> (30 pts) Approx. 3 days	
This unit starts with an introduction to arrays. Here you'll practice making arrays of different types and different sizes. In addition, you'll use them to store information for each of the Java Tasks for this part of the unit. We'll also look further at codingbat.com as we continue to tackle methods that work with Strings in different ways.	 Array Notes
	 Complete 8 Array Tasks
	 8 Coding Bat Challenges
	 Check-off from Mr. Benshoof
<b>Part 2: Multi-dimensional Arrays</b> (40 pts) Approx. 3 days	
Here in the second part of this unit, we'll get into the details of using multi-dimensional arrays. Multidimensional arrays are super useful when we need to keep track of data with multiple variables, or data that can be envisioned like a grid. Here you'll learn how to instantiate and work with multidimensional (mostly 2D) arrays, as well as how to use a nested for loop to work with them. We'll also continue working with codingbat.com and some great String challenges.	 2D Array Notes
	 Complete 6 2D Array Tasks
	 8 Coding Bat Challenges
	 Take the Unit 5 Quiz!
	 Check-off from Mr. Benshoof
<b>Part 3: Tic-Tac-Toe</b> (30 pts) Approx. 2 days	
The final part of our unit will have you apply 2D arrays in the creation of a small tic-tac-toe game. You'll be able to use your 2D array to store the values of the player 'X' and 'O' tokens as well as use a nested for loop to traverse the grid and display it properly back to the user. You might even make the game playable by the computer and build in some simple AI decision making!	 Tic-Tac-Toe Brainstorm Notes
	 Create Your Tic-Tac-Toe Game
	 Write Log 5: Arrays
	 Check-off from Mr. Benshoof







(30 pts) Approx. 3 days

This unit covers the creation and use of arrays. *Arrays* are lists of values that get stored with a single name. For example, if you had a list of prime numbers, you might make an array of them as shown below:

```
int[] primes = {2, 3, 5, 7, 11, 13, 17, 19}
```

In this instantiation, the square brackets make the variable a list (array) of values rather than one single one. The numbers in the curly braces are automatically placed into the array, and the array automatically (and forever) has size 8 because that was how many numbers we gave it. Later, if I wanted to recall the '11', I can reference `primes[4]` and use the value returned by the array. Arrays will be super useful as we start trying to store more and more information!

1. Take a full page of good notes on 1-dimensional arrays as you watch the three presentations from our website. Make sure that your notes include good details on the different ways to make arrays, how to store information in arrays, how to get information from arrays, and how to use a for loop to traverse arrays!
2. Now, complete the next 8 Java Tasks below that are all about defining classes, their variables, and their constructors!
  - a. JAVA TASK 38: Make a program that simply instantiates 4 different arrays (in either different ways or storing different kinds of information). Make a for loop print the array nicely on the screen.
  - b. JAVA TASK 39: Make a program that creates an array of Strings that is 4-spaces long, but empty. Then, let the user enter 4 different words, and add each of them to the array. Then, use a for loop to print the array on the screen.
  - c. JAVA TASK 40: Make a program that creates an array of 9 random double values. Use a for loop to print the array nicely on the screen.
  - d. JAVA TASK 41: Make a program that creates an array of 9 random integers between the values 5 and 9 inclusive. Use a for loop to print the array nicely on the screen.
  - e. JAVA TASK 42: Make a program that creates an array of 6 boolean values that alternate 'true' – 'false'. Use a for loop to print the array nicely on the screen.
  - f. JAVA TASK 43: Make an array that shows the results of 16 coin flips. The array should be filled with Strings that are either "H" or "T". The values should be generated randomly by the program and should have a 50% chance of being heads and a 50% chance of being tails. Use a for loop to print the array on the screen.
  - g. JAVA TASK 44: Make a program that creates an array of 15 integers. Use a for loop to fill the array with increasing numbers from 1 to 15. Use another for loop to print the array nicely.
  - h. JAVA TASK 45: Make a program that creates an array of 50 integers. Use a for loop to fill the array with the first 50 numbers of the Fibonacci sequence. Then, use another for loop to print the array nicely.
3. Finally, return to codingbat.com and log in with your email address. Look for the "String-2" group of problems. Pick and solve any 8 of these challenges. These string challenges will need us to use the `substring()` method as well as a for loop to search through larger strings.

Part 1: Tasks	10-8 points	7-4 points	3-0 points
 Array Notes	+ Watch all presentations + You took a full page of notes on 1-dimensional arrays + Your notes include details about instantiation, editing, and traversing arrays	- Less than a full page of notes on arrays - Your notes are missing important parts	- Very brief or no notes in your notebook
 Java Tasks 38-45	+ You completed all 8 Java Tasks from this section	- You did not complete all 8 tasks	- You did not complete any tasks
 Coding Bat	+ You completed 8 String#3 challenges in codingbat.com	- You completed 7 - 5 codingbat.com challenges	- you completed fewer than 5 codingbat challenges
 Checkoff from Benshoof	+ Mr. Benshoof got to see your array programs run successfully	N/A	N/A







(40 pts) Approx. 3 days

In addition to being great ways to store **lists** of information, arrays are also great ways to store **grids** of information! To use arrays to store a grid, we simply need to make an array *full of other arrays!* This is – in many ways – like a nested for loop, where one structure is storing similar versions of itself. Here, we’ll need to learn to create 2D arrays, and it’ll be as simple as adding another pair of square brackets ( `[]` ):

```
int[][] gridOfNums = new int[][]
```

In the instantiation above, we have to double-up on the pair of brackets. The first represent the rows of the grid, while the second represents the columns. In this way, we can assign and address individual spots in the grid by referencing `gridOfNums[rowSpot][colSpot]`.

1. Watch the three videos on *2D Array Overview*, *More about 2D Arrays*, and *2D Arrays of Objects*. Take a full page of good notes on these topics. Make sure your notes include details about how to make 2D arrays, how to assign values to them, and how to use a nested for loop to traverse these new arrays!
2. Now, complete the following 6 challenges that all require you to create and work with 2-D Arrays.
  - a. JAVA TASK 46: Make a program that simply instantiates a 2D array of consecutive integers that is 5 rows and 7 columns (5x7) big. Use a nested for loop to print the array like a nice grid on the screen.
  - b. JAVA TASK 47: Make a program that creates a 2D array of Strings that is 6x6. These spots should then ALL be filled with the String “LHS”. Use a nested for loop to print the array like a nice grid on the screen.
  - c. JAVA TASK 48: Make a program that creates a 2D array of 0’s and 1’s, and uses the `Math.random()` method to assign each spot to either a 0 or 1 randomly. Print the array with a nested for loop!
  - d. JAVA TASK 49: Make a program that creates a 2D array of Strings that is 3x3. Let the user type in 9 different words and put those words into the array. Print the array with a nested for loop!
  - e. JAVA TASK 50: Write a program that creates a 2D array of integers that is 6x6 and filled with 0’s. Print the array on the screen. **Then**, let the user enter a new integer into the array, and have them choose which location their new integer should go. The program should replace that value in its array, and then print the array on the screen again!
  - f. JAVA TASK 51: Write a program that makes a 5x5 2D array of strings, where the center space is an “X” and the other spaces are all “O’s”. Then, let the user enter commands like ‘up’, ‘down’, ‘left’, or ‘right’ and have those commands change the location of the ‘X’, allowing them to walk the X around the grid.
3. Lastly, go back to CodingBat.com and log in with the e-mail you provided when you made the account. Go to the Java resources and the “String 2” section. Pick any 8 challenges from the String 2 group and complete them!




Part 2: Tasks	10-7 points	6-4 points	3-0 points
 2D Array Notes	+ Watch all presentations on 2D Arrays + Take a full page of notes on 2D Arrays and how to use them	- Less than a full page of methods notes - No notes on 2D Arrays	- Very brief or no notes in your notebook
 Java Tasks 46-51	+ You completed all 6 Java Tasks from this section	- You did not complete all 6 tasks	- You did not complete any tasks
 Coding Bat	+ You completed 6 String #2 challenges in codingbat.com	- You only completed 4 or 5 codingbat.com challenges	- you completed fewer than 4 codingbat challenges
 Take the Unit 5 Quiz!	+ You took the Unit 5 Quiz by the due date + Your score is based on the number you get correct	N/A	- You did not take the Unit 5 Quiz



(30 pts) Approx. 2 days

The last part of this unit will ask you to combine a wide variety of things we know about programming with our new knowledge of 2D Arrays to create a simple Tic-Tac-Toe game. Here you'll need to think about the different parts your program will need to have, and how you'll show your tic-tac-toe grid to the user. In addition, you'll need to keep track of which players have played in each space and keep the game updated!

1. Start – as usual – by brainstorming what you might need in order to make a tic-tac-toe game. What classes might you want? (Not that you need multiple classes). How will you show the grid on the screen? How will you take user input? What kind of 2D array do you want to manage the data of the game?
2. Create your program such that:
  - a. Two players can play tic-tac-toe against each other
  - b. Users should be able to take enter their moves by taking turns at the keyboard
  - c. The game keeps track of who has played in each space
  - d. The game displays the updated grid after every person's move
  - e. You **might** want to make your program prevent players from taking spots that are already occupied
  - f. You **might** want to make your program check to see if someone has won the game!
  - g. Your program should be entirely text-based... no need to try and make graphics other than letters on the screen!
3. *Log 5: Arrays* - The last part of our unit is to write a full-page response about arrays of all kinds. What uses can you anticipate for arrays? How might arrays of different kinds be useful? What's trickiest about working with arrays and 2D arrays? What important information about arrays do you want to make sure you remember in the future?

<b>Part 3: Tasks</b>	<b>5 points</b>	<b>4-3 points</b>	<b>2-1-0 points</b>
 Tic-Tac-Toe Brainstorm Notes	+ You wrote a full page of brainstorms, ideas, and notes about how to structure your program	- You wrote less than a page - Your notes do not outline a coherent plan	- Your notes are lacking or missing - There is no plan for your program
	<b>15-10 points</b>	<b>9-5 points</b>	<b>4-0 points</b>
 Create Your Tic-Tac-Toe Program	+ You created your Tic-Tac-Toe program according to the criteria listed above. + Your Tic-Tac-Toe Program works as intended + Play a game of Tic-Tac-Toe with Mr. Benshoof!	- Your Tic-Tac-Toe program does not meet some criteria - Your Tic-Tac-Toe program mostly works	- Your Tic-Tac-Toe program does not work at all
	<b>10-8 points</b>	<b>7-4 points</b>	<b>3-0 points</b>
 Log 5: Arrays	+ You wrote a complete page in your engineering notebook	- You wrote less than a full page	- You wrote less than half a page

