

Topic: Introduction to Arrays in C (*For non-programmers, an array is a set of values under a single identifier whose members are identified by an index number. For example score[0], score[1], score[2], etc.*)

Summary: Students will be introduced to arrays in C. Students will learn to create, populate, and output the values in an array.

Context & Content: This is the first day of the arrays unit in an semi-independent study programming class of three students (10th grader, 11th grader, & 12th grader).

Context in Larger Unit: The students were introduced to the C programming language in the last unit and know how to do basic input and output, store values in variables of various types, use conditional statements and loops, and use functions to enhance readability. In the introductory unit, they learned about representations of digital data.

Michigan Computer Science Standards:

- 3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
- 3A-AP-15: Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.

Objectives (framed in terms of what the student should know or be able to do before the next lesson):

- Students will be able to explain how array items are named and referenced.
- Students will be able to create an array of a given length and populate it with values.
- Students will be able to store a value at any position in an array.
- Students will be able to iterate through an array and display its contents.

Technology Resources:

- [Slide deck for the day](#): Students have an exact copy of this Google Slides document that acts as a hyperdoc.
- [CS50 Week 2 Resources](#): CS50 is a Harvard University class that is offered freely online through the edX portal. There are videos, notes with screen captures and content explanations, supplemental videos, and an active social media presence for support.
- [Class Website](#): The website acts as a portal for the class. Students have access to all of the resources for the class on CS50, supplemental notes from me (in development), and the slide deck of the day.
- [GitHub CodeSpace](#) These are custom programming environments created by CS50

Assessment: Students will have an oral formative assessment at the beginning of the class when the students do a self-review of the concepts from the previous class. Students will have two visual assessments during the class where I look at their code to make sure they are progressing. Finally, students will have a summative assessment on the content and skills presented when I grade their code (due the next day).

Lesson Flow	Assessments and Assignments
<u>Opening Tasks</u> (5 min): • Students will review the skills from the prior lesson.	• Formative Assessment: oral review of the concepts

- Students will be given the following scenario as a warm-up exercise: “You work as a teacher and need to keep track of the students’ average on three quizzes. How would you do it? Write it up in pseudocode. How would you handle more quizzes (i.e. the 4th, 5th, 6th, etc.)? Share with your row.”

Teaching:

- Discuss as a class the various plans to handle the warm-up problem. Focus on solutions for adding more quizzes. Steer the conversation towards the awkwardness of variables like “score1,” “score2,” “score3,” etc. (2 min)
- Demonstrate for the students how to activate the Magnifier accessibility tool in Windows. (2 min)
- Walk through the code on slide four and steer the conversation to the point above. (3 min)
- Introduce the concept of an array and go through the code on slide five. (3 min)
- Go through the Goals slide. (2 min)
- Have students create the program on slide seven. Walk around asking leading questions and pointing students to the resources if they are stuck. Encourage struggle. (7-10 min)
- On slide eight, students will analyze the code and look for possible problems, especially as the number of quizzes increases. Direct discussions to discover that we are back to the original problem with multiple variables only with an array this time. (3 min)
- On slide nine, students will compare two code snippets analyzing the new code in light of the old code's deficiencies. Students should see if they can figure out how the new code works. (3 min)
- On slide ten, show students how the new code allows for dynamic use. Direct students to see how arrays and FOR loops work well together. (2 min)
- Quickly introduce students to an application of arrays-- encryption. Tell them that one of the problem set problems will be to create a cipher. (2 min)
- Slide twelve is homework. Students will use what they have learned and the models they have seen to make a working program which dynamically takes any number of quiz inputs, prints them out as a list, and calculates the average. (Remaining time)

- Formative Assessment: visually assess that students are correctly coding.
- Formative Assessment: visually assess that students are correctly coding.
- Summative Assessment: Students will produce the program to the left (slide 10) and turn in to Google Classroom.

Closure (Did we accomplish what we set out to do today?):

- ❖ Ask, did we accomplish our goals for today?
- ❖ Review and check off objectives.