



# DATA STRUCTURES LESSON

Dan Bonarigo, Riverside Brookfield High School, Riverside, IL



This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.

Additional materials may be found at [www.ncyte.net](http://www.ncyte.net)



# DATA STRUCTURES

This module provides exposure to data structures in the form of lists.

## OVERVIEW

**Lesson Description:** This module provides exposure to data structures in the form of lists.

**Prerequisite Knowledge:** Students should have been introduced to strings, Booleans, user input, variables, conditionals and iteration.

**Length of Completion:** This lesson should be completed in 4-5 hours/class periods.

**Learning Setting:** This lesson is intended for a face-to-face learning environment.

**Lab Environment:** This CCL is designed to be taught in a classroom which provides access to Internet-connected computers containing an Integrated Development Environment (IDE) consistent with class' chosen programming language.

**Activity/Lab Tasks:** Students will write a program or programs that implement the use of data structures, For and/or While loops and conditional statements involving variables, Booleans, user input and strings.

01.DataStructures\_Overview.docx

02.DataStructures\_Presentation.pptx

03.DataStructures\_CodeSegmentsandOutput\_Activity.docx

04.DataStructures\_CodeSegmentsandOutput\_ActivitySolutions.docx

05.DataStructures\_CodeSegmentsandOutput\_ActivityTwo.docx

06.DataStructures\_CodeSegmentsandOutput\_ActivityTwoSolutions.docx

07.DataStructures\_UsernameProgramming\_Activity.docx

08.DataStructures\_UsernameProgramming\_ActivitySolutions.docx

## LEARNING OBJECTIVES AND AP CSP ALIGNMENT



## Lesson Learning Objectives

Students will:

1. Develop a program to validate input,
2. Test whether inputs to a program are producing the expected output, and
3. Represent data in various types, including numbers, strings, lists, and Booleans.

### ASSOCIATED AP CSP SUB LEARNING OBJECTIVES

#### AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 1: CREATIVE DEVELOPMENT

- LO CRD-2.B Explain how a program or code segment functions.
  - CRD-2.B.1 A program is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as software.
  - CRD-2.B.2 A code segment refers to a collection of program statements that are part of a program.
  - CRD-2.B.3 A program needs to work for a variety of inputs and situations.
  - CRD-2.B.4 The behavior of a program is how a program functions during execution and is often described by how a user interacts with it.
- LO CRD-2.C Students will be able to identify inputs in a program.
  - CRD-2.C.1 Program input is data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text.
  - CRD-2.C.4 Inputs usually affect the output produced by a program.
  - CRD-2.C.6 Input can come from a user or other programs.
- LO CRD-2.D Students will be able to identify outputs in a program.
  - CRD-2.D.1 Program output is any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text.
  - CRD-2.D.2 Program output is usually based on a program's input or prior state (e.g. internal values).



- LO CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.
  - CRD-2.J.1 In the development process, testing uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.
  - CRD 2.J.2: Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data.

---

### AP COMPUTER SCIENCE PRINCIPLES COURSE, BIG IDEA 3: ALGORITHMS AND PROGRAMMING

- LO AAP-1.A Represent a value with a variable.
  - AAP-1.A.1 A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values.
  - AAP-1.A.2 Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables.
  - AAP-1.A.3 Some programming languages provide types to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings.
- LO APP-1.B Determine the value of a variable as a result of an assignment.
  - AAP-1.B.1 The assignment operator allows a program to change the value represented by a variable.
  - AAP-1.B.3 Conditional statements or “if-statements” affect the sequential flow of control by executing different statements based on the value of a Boolean expression.
- LO AAP-1.C Represent a list or string using a variable.
  - AAP-1.C.4 A string is an ordered sequence of characters.
- LO AAP-2.E For relationships between two variables, expressions, or values:
  - Write expressions using relational operators
  - Evaluate expressions that use relational operators
  - AAP 2.E.1 A Boolean value is either true or false
- LO AAP-2.H Students will be able to write and evaluate conditional statements.



- AAP-2.H.1 Conditional statements or “if-statements” affect the sequential flow of control by executing different statements based on the value of a Boolean expression.
- LO AAP-2.K For iteration:
  - Write iteration statements
  - Determine the result or side effect of iteration statements.
- AAP-2.K.1 Iteration statements change the sequential flow of control by repeating a set of statements zero or more times until a stopping condition is met.
- LO AAP-2.N For list operations:
  - Write expressions that use list indexing and list procedures.
  - Evaluate expressions that use list indexing and list procedures.
- APP-2.N.2: List procedures are implemented in accordance with the syntax rules of the language.
- LO AAP 2.O For algorithms involving elements of a list:
  - Represent using iterative statements to traverse a list.
  - Determine the result of an algorithm with list traversals.
- AAP-2.O.1: Traversing a list can be a complete traversal where all elements in the list are accessed, or a partial traversal where only a portion of elements are accessed.

## LESSON DETAILS

**Overview of Lessons:** The teacher should use the presentation to review and introduce the content and then allow students to complete the activities.

Lesson 1:

- Presentation (Slides 4-17)

Lesson 2:

- Presentation (Slides 18-23)
- Code Segments and Output Activity

Lesson 3:

- Presentation (Slides 24-29)
- Code Segment and Output Activity Two

Lesson 4:



- Presentation (Slides 30-33)
- Username Programming Activity

## LESSON 1

### Presentation

02.DataStructures\_Presentation.pptx

Slides 4-17 Slides 4-17 are used to review Booleans, strings, user input, variables, conditionals and loops (teachers may choose to skip any review slides as needed).

## LESSON 2

### Presentation

02.DataStructures\_Presentation.pptx

Slide 18-23 introduces lists and list operations: once these slides are completed, students should start Code Segments and Output Activity.

### Code Segments and Output Activity

03.DataStructures\_CodeSegmentsandOutput\_Activity.docx

04.DataStructures\_CodeSegmentsandOutput\_ActivitySolutions.docx

Students should complete Code Segments and Output Activity immediately following instruction of slide 23. This worksheet is intended to give students an opportunity to verify their understanding of lists. Students should turn in their paper for grading and/or verification of completion. Teachers should use this as an opportunity for formative assessment to verify students correct understanding of lists.

## LESSON 3

### Presentation

02.DataStructures\_Presentation.pptx

Teachers should continue with the presentation using slides 24-29 to introduce the FOR EACH loop with lists and then students should complete Code Segments With Error Activity.

### Code Segments and Output Activity Two



05.DataStructures\_CodeSegmentsandOutput\_ActivityTwo.docx

06.DataStructures\_CodeSegmentsandOutput\_ActivityTwoSolutions.docx

This worksheet is intended to give students an opportunity to verify their understanding of the FOR EACH loop with lists. Students should turn in their paper for grading and/or verification of completion. Teachers should use this as an opportunity for formative assessment to verify students correct understanding of FOR EACH loops with lists.

## LESSON 4

### Presentation

02.DataStructures\_Presentation.pptx

Teachers will conclude the lesson with the presentation slides 30-33 and then students will complete the Username Programming Activity.

### Username Programming Activity

07.DataStructures\_UsernameProgramming\_Activity.docx

08.DataStructures\_UsernameProgramming\_ActivitySolutions.docx

This worksheet is intended to give students an opportunity to write programs using lists and FOR EACH/REPEAT loops. This activity also exposes students to some basics of username/password security and username/password creation. Although sample programs are given, teachers should help each student use their creativity to arrive at a unique solution to the programming tasks. Example solutions are provided in the solutions document. Teachers should use this as an opportunity for formative assessment to verify students' ability to write a program using lists and loops.

The teacher should guide a discussion on data breaches and the ethical implications while presenting slide 34.

## ACKNOWLEDGEMENTS

### Resources:

[2020-21 Updates To AP Computer Science Principles | AP Central—The College Board. \(2018, November 6\) PDF.](#)



This document is licensed with a [Creative Commons Attribution 4.0 International License](#) ©2017 [Catalyzing Computing and Cybersecurity in Community Colleges](#) (C5). This material is based upon work supported by the **National Science Foundation** under Grant No.1548315.