Unit 3: Standard Data Structures

Content Area:	**Computer Science**
Course(s):	
Time Period:	**Marking Period 1**
Length:	**3-4 Weeks**
Status:	**Published**

Summary

Students will represent information within a program by using data structures. They will choose appropriate data structures based upon the need within the program, taking care to note storage use. They will use classes on/or objects to assist in the writing of programs/code.

Revised Date: July 2025

MA.9-12.1.2.12prof.Cr	Creating
MATH.K-12.1	Make sense of problems and persevere in solving them
ELA.L.SS.11–12.1	Demonstrate command of the system and structure of the English language when writing or speaking.
MATH.K-12.2	Reason abstractly and quantitatively
ELA.L.KL.11–12.2	Apply knowledge of language to understand how language functions in different contexts, to make effective choices for meaning or style, and to comprehend more fully when reading or listening.
MATH.K-12.4	Model with mathematics
MATH.K-12.5	Use appropriate tools strategically
MA.9-12.1.2.12prof.Cr2	Organizing and developing ideas.
MATH.K-12.6	Attend to precision
MATH.K-12.7	Look for and make use of structure
MATH.K-12.8	Look for and express regularity in repeated reasoning
CS.9-12.8.1.12.AP.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
CS.9-12.8.1.12.DA.2	Describe the trade-offs in how and where data is organized and stored.
CS.9-12.8.1.12.DA.4	Explain the relationship between binary numbers and the storage and use of data in a computing device.
WRK.K-12.P.5	Utilize critical thinking to make sense of problems and persevere in solving them.
WRK.K-12.P.8	Use technology to enhance productivity increase collaboration and communicate effectively.
TECH.9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
TECH.9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
	Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.

Essential Questions / Enduring Understandings

Essential Questions:

- When do you use each of the data structures to represent information in a program?
- How do you manipulate the date structures to hold data?
- What are simple data types?
- When would you use a given simple data type?
- What is an abstract data type?
- What is a data structure?
- How do you choose a datas structure?
- What are the operations that can be used on data structures?
- Given an operation on data structures, explain how can a given operation on data structures be implemented into a program?
- How are dimensional arrays used in programs?
- How is a class utilized in a program?

Enduring Understandings:

- Problems need to be solved in order to meet the needs of the user.
- Abstract data types and operations can be used to solve problems.
- Classes can be used to simplify coding.
- One-dimensional arrays can be used to store data.
- Different data types are used for different tasks.

Objectives

Student will know:

- when to use each of the simple data types
- the purpose behind classes
- how to implent a one-dimensional array
- how to include build-in or library functions and structures
- how to design a user interface
- how to choose test data
- how to debug a program

Students will be skilled at:

- recognizing data structures, their storage capabilities and when to use each type

Learning Plan

- Preview the essential questions and connect to learning throughout the unit.
- Discuss the use of primitive data types and when best to use them.
- Write programs that implement primitive data types to a given set of specifications.
- Discuss abstract data types and proved examples of when to use them.
- Using a series of problems, discuss with the students the different operations that will be required in order to solve the problem.
- Discussion of classes.
- Discussion of accessor methods for classes.
- Discussion of interactions of classes.
- Identify and implement class structures.
- Use of strings in programs.
- Discussion of how arrays are created.
- Use of one-dimensional arrays.
- Discuss the different operations that can be used on data structures.
- Implement a given operation on a data structure by writing a program that meets certain specifications.
- Have students work with a variety of data structures in developing programs that meet certain specifications.
- Have students write programs to a given set of specifications and then implement a variety of tests upon the program.

Assessment

- Assessments
    - Formative: Daily assessments using examples from class notes and CodeHS.com, AP Classroom/Albert Checks for Understanding
    - Summative: Teacher-created assessments/projects and CodeHS Computer Science Projects, AP Classroom/Albert Unit Assessments
    - Benchmark: Check for understanding benchmark assessments on CodeHS, AP Classroom/Albert/Khan Academy Diagnostics
    - Alternative Assessments: Student-centered activities such as a doorbell coding project, game design projects, and other activities involving real world applications
        - Complete quizzes/test: Algorithms, Structure of Programs, Design of Programs
        - Be observed by the teacher during individual work on the performance tasks.

    - Conduct self-assessments and reflections
    - Conduct Peer Evaluations.

Materials

- Core instructional materials: Core Book List
- Supplemental materials: CodeHS, computers, and reference books.

Integrated Accommodations and Modifications

See Linked Document.