Unit 3: Data Types-Arrays, Enums and Structs

Content Area: **Technology**
Course(s): **Computer Programming C++ I & II**
Time Period: **8 weeks**
Length: **Weeks**
Status: **Published**

Unit Overview

Students will be able to read with understanding and write programs involving arrays, enumerated types (enums) and compound data structures (structs). Students will be able to take a description of a problem and define data types appropriate to the problem, and write code that uses these data types in algorithms. Topics covered include arrays of primitive types (int, float, char), iterating over arrays using for-loops, linear search of arrays, c-style strings, the sizeof operator, declaring and using enums, declaring and using structs, and nested data structures.

Transfer

Students will be able to independently use the knowledge acquired in this unit to read and write C++ programs making use of user-defined data types appropriate to the problem at hand. Students will understand what arrays, enumerations and structures are, when they are useful and appropriate to a particular problem, and how they are used. Students will be able to analyze a given problem in terms of the data types needed, declare the data types with correct syntax, and write code that makes use of the structures, using common code idioms.

Meaning

Understandings

Students will understand that...

-arrays are used to hold many pieces of data of a single type, and array elements are accessed using the appropriate index

-enumerated types (enums) are used to represent variables with a finite number of known possibilities (e.g. On/Off, Up/Down/Left/Right), and can be converted to integers

-compound data structures (structs) are used to bind together related data. Structs can hold any number of pieces of data of any type

-structs, enums, and arrays can be nested: e.g. a struct can hold an array whose elements are of an enumerated type, etc...

-like all information, data is easier to understand when organized into a coherent plan. C++ provides arrays,

enumerated types, and compound data structures to organize data.

Essential Questions

Students will keep considering...

-Under what circumstances are arrays, enumerated types, and structs appropriate to a problem at hand?

-What is the best organizational scheme for the data or information implicit in a given problem description?

-What algorithms are appropriate for accessing and modifying the contents of arrays and structs?

-What are the differences between arrays and structs in terms of what kind of data they can store and how that data is accessed?

-What are the caveats when converting between enumerated types and integers (enums and ints)?

-How can I tell what the valid indices are for an array and ensure that no out-of-bounds errors occur?

-What is the relationship between the number of elements in an array and the maximum valid index for an array?

Application of Knowledge and Skill

Students will know...

-arrays are declared by naming the type of the element, the name of the array, and the number of elements in square brackets

-array elements are accessed using square brackets and an index

-arrays can be initialized either with explicit setting of each element, with a for loop, or with an initializer list

-valid indices for an array run from 0 up to 1 less than the number of elements

-for loops are a convenient way to iterate over the elements of an array

-the sizeof operator can be used to determine the number of bytes in memory used by an array

-enumerated types (enums) are declared using the keyword enum, the name of the new data type, and a comma-separated list of named possible values in curly braces

-enums can be freely converted to ints, but the reverse requires an explicit cast

-compound data structures (structs) are declared with the struct keyword, the name of the new data type, and a semicolon-separated list of data members (each with a type and a name) enclosed in curly braces

-struct elements are accessed using the dot operator and the name of the data member

-struct variables can be initialized with an initializer list or by explicitly assigning the value of each data member

-structs may contain enums, arrays of any data type, and other structs (nesting); arrays can be of any data type, including a struct

Students will be skilled at...

-declaring arrays and accessing and modifying their elements

-using bounds checks to ensure array indices are not out of bounds

-initializing arrays by various methods

-declaring and using enumerated types, and converting enumerated types to and from integral types (ints)

-defining, declaring, initializing and using structs

-constructing nested data structures consisting of structs, enums and arrays

-determining when various ways to organize data are appropriate

-reading and understanding other's code containing custom data types

Academic Vocabulary

type cast, compound data type, data structure, defining vs. declaring, array, element, index/indices, enumerated type, c-string, initializer list, data member, reserved word

Learning Goal 1

Students will be able to read with understanding and write code using arrays of primitive data types and enumerated types, and contrast the types of problems that can be solved with arrays versus those that are solved with ordinary variables.

Target 1

Students will be able to declare array variables and use them to solve programming problems, and use models to illustrate the layout of arrays in memory.

Additional standards:

9.3.IT-PRG.1, 2,3

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.2	Analyze the relationships between internal and external computer components.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).

Target 2

Students will be able to define enumerated types and use them to solve programming problems involving systems with a finite set of states, predict the results of programs that use them.

Additional standards:

9.3.IT-PRG.4,5,6

TECH.8.2.12.E	Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Learning Goal 2

Students will be able to read with understanding and write code using compound data types (structs), and identify the situations where structs are applicable, and solve programming problems with structs.

Target 1

Students will be able to solve programming problems using compound data structures (structs) containing primitive data types.

Additional Standards:

9.3.ST.6, 9.3.ST-ET.1, 3

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Target 2

Students will be able to write code that solves programming problems using structs containing arbitrary data types (arrays and enumerated types) and nested data structures, and identifying programming problems where nested structs are useful.

Additional Standards:

9.3.ST-ET.5, 9.3.IT.2

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Learning Goal 3

Students will investigate career opportunities in computing and programming fields and analyze the impact computing has on society.

Target 1

Students will analyze the impact of technology in the information age on the types of job opportunities available.

Additional Standards:

9.3.ST5, 9.3.ST-SM.3

CAEP.9.2.12.C.1	Review career goals and determine steps necessary for attainment.
CAEP.9.2.12.C.2	Modify Personalized Student Learning Plans to support declared career goals.
CAEP.9.2.12.C.3	Identify transferable career skills and design alternate career plans.

Target 2

Students will analyze the impact computing has on society in both the United States and in developing countries.

Additional Standards:

9.3.ST-ET.2

TECH.8.2.12.D.4	Assess the impacts of emerging technologies on developing countries.
TECH.8.2.12.D.6	Synthesize data, analyze trends and draw conclusions regarding the effect of a technology on the individual, society, or the environment and publish conclusions.

Summative Assessment

-Chapter and benchmark assessments

-Project-based assessments. Possible projects include:

-Gradebook manager

-Contact List

-Scavenger Hunt

-Baseball Simulater

-Bokmon (Pac-Man clone)

21st Century Life and Careers

CAEP.9.2.12.C.1	Review career goals and determine steps necessary for attainment.
CAEP.9.2.12.C.2	Modify Personalized Student Learning Plans to support declared career goals.

CAEP.9.2.12.C.3	Identify transferable career skills and design alternate career plans.
CAEP.9.2.12.C.4	Analyze how economic conditions and societal changes influence employment trends and future education.
CAEP.9.2.12.C.7	Examine the professional, legal, and ethical responsibilities for both employers and employees in the global workplace.

Formative Assessment and Performance Opportunities

Daily challenge tasks, individual and group project presentations, portfolio assessments, and in-class observation, demonstration and questioning.

Accommodations/Modifications

-Supplemental project guides

-One-on-one assistance

-Additional challenges (possible examples):

-Add randomness and additional goals to the scavenger hunt project

-Make the baseball simulation more realistic and search for optimal batting orders

-Enhance the Deli Counter project with more choices for food, etc

Unit Resources

Project guides (e.g. baseball simulation, scavenger hunt, Bokmon/Pac-Man clone)

Youtube videos on jobs in computer fields

Interdisciplinary Connections

MA.K-12.4	Model with mathematics.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
LA.RST.11-12.4	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11-12 texts and topics.
LA.RST.11-12.7	Integrate and evaluate multiple sources of information presented in diverse formats and

media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.