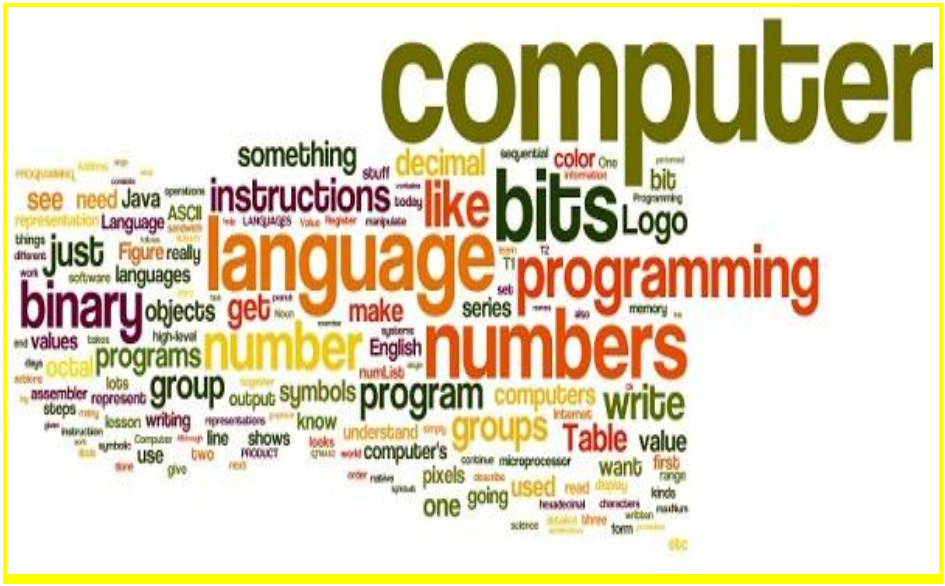
COMPUTER PROGRAMMING (WITH MATHEMATICAL MODELING) CURRICULUM





Grade Level(s): 10-12

Curriculum Author(s): Raymond Robillard

Course Description: Computer Programming (with Mathematical Modeling) is a semester-long course that offers an introduction to the theory and practice of developing computer programs with an emphasis on techniques of program development steeped in object-oriented thinking. Many of the programs will be geared toward mathematical processes, and thus, this course will include a heavy math component. Students will explore topics such as control structures, input/output, loops, objects, classes, simple data structures, and basic concepts of software development. The programming language used in the course is JAVA.

Year At A Glance			
Unit Title	**Overarching Essential Question**	**Overarching Enduring Understanding**	**Vision of A Learner "I Can" Statements**
Introduction to Number Systems	• Are there other ways to count and represent numbers?	• Value can be represented in a limitless number of ways, each based on a unique base number system, but the most commonly used in Computer Science are binary, octal, decimal, and hexadecimal.	TCC1(9-12), TCC3(9-12), AA1(9-12)
Working with Types of Data	• How is information input to, stored in, and output by a computing device?	• Information is input to a computing device, abstracted to a binary representation, and then output back in its original form.	TCC3(9-12), TCC4(9-12), TI3(9-12)
Branches and Loops	• How does a computer program use selection and iteration to manage code complexity?	• Control structures, such as conditional statements and loops are abstractions that enable increased complexity to computer programs	TCC4(9-12), TI3(9-12), P4(9-12)
Advanced Programming Techniques	• How do programming languages change and evolve over time?	• Common procedures, notations, and structures have been simplified and improved to meet the needs of Java programmers.	TCC3(9-12), AA1(9-12), AA4(9-12)
Arrays	• How do arrays, and other collection objects, manage code complexity?	• An Array is a data collection tool that allows for organization of real world lists such as rosters, statistics, testing results, etc. using indexes and a series of common searching and sorting algorithms.	CCE2(9-12),. TI3(9-12), P4(9-12)

Unit 1 - Introduction to Number Systems

Desired Results - Goals, Transfer, Meaning, Acquisition

Established Goals:
1A-AP-09 - Model the way programs store and manipulate data by using numbers or other symbols to represent information.
1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
2-DA-07 - Represent data using multiple encoding schemes.
3A-CS-01 - Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
3A-DA-09 - Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.
3B-CS-02 - Illustrate ways computing systems implement logic, input, and output through hardware components.

Vision of A Learner Attributes: Students will be able to independently use their learning to... ("I can" statements to be demonstrated)
TCC1(9-12): I can ask purposeful, insightful questions to find a variety of innovative solutions.
TCC3(9-12): I can integrate relevant information to produce multiple valid solutions.
AA1(9-12): I can evaluate different approaches and justify the best pathway to success.

Understandings: Students will understand that...
- Value can be represented in a limitless number of ways, each based on a unique base number system, but the most commonly used in Computer Science are binary, octal, decimal, and hexadecimal.
- A place value represents a quantity of some power of the specified base.
- Language and Images can be abstracted down to 0's and 1's.

Essential Questions:
- Are there other ways to count and represent numbers?
- How do place value and numerical symbols represent value?
- Why do computers use a binary rather than a decimal number system?

Students will know…
- The place values of binary, octal, decimal, and hexadecimal number systems are powers of 2, 8, 10, and 16 respectively.
- The letters "A" through "F" represent digits in a hexadecimal system.
- The smallest base that a number could possibly be in is one more than the largest individual digit.

Students will be able to...
- Convert between any two base number systems.
- Approximate decimal values with negative powers of an indicated base.
- Translate characters and colors between binary and ASCII values of RGB values.

• Characters of text are represented in binary using the Ascii system. • Pixels of an image are represented using the RGB color model	

Key Vocabulary: Number System, Binary, Octal, Decimal, Hexadecimal, Place Value, Base, Power, ASCII, Character, Pixel, RGB

Assessment Evidence

Performance Tasks: **Summative Unit Assessment TCC1(9-12), TCC3(9-12), and AA1(9-12)** Topics will include: • Converting numbers from decimal to binary, octal, and hexadecimal. • Converting numbers from binary, octal, and hexadecimal to decimal. • Explaining how text is represented in binary form • Explaining how color is represented in binary form	**Other Evidence:** Classwork tasks described below • Base Conversion Worksheet • Text Conversion Worksheet Quiz on converting between various number systems

Learning Plan

In its most primitive form, all communication began at a binary level. Yes/No, On/Off, Black/White, etc. are all abstractions of more complex concepts. But, because a computer behaves strictly on the presence, or lack of, electric impulses, they must reduce all information to this binary system. In this unit, students learn how all data can be represented in a binary form, and how we can translate back and forth between these and other forms.

- Binary Learning Activity- Students will be asked to consider a world in which we don't have fingers, but merely stubs on each arm. In this hypothetical world, we can't learn to count in the same way as we haven using our fingers. How does this change things? Students will come to see that regrouping after counting to 2 is the new norm, rather than counting to 10. And in this world, we may have only devised symbols to represent two digits, 0 and 1. This class-long activity includes discussion that focuses on place value, symbolic representation with digits, and other possible base systems(octal and hexadecimal) that would arise in a society of people that have 8 or 16 total fingers. **TCC1(9-12)**
- Base Conversion Worksheet- Students will practice their learning through practice problems that require them to convert between various bases.**AA1(9-12)**
- Text Conversion Activity- Students will be provided a copy of an ASCII conversion chart (https://www.asciitable.com/) and research the origins and the history of the system. Students will discuss implications for ASCII given the numerous languages throughout the world, including some that use different characters in their alphabet, and how these differences in language could be addressed.**TCC3(9-12)**
- Text Conversion Worksheet- Students will be tasked with converting back and forth between written text and ASCII code.**AA1(9-12)**
- RGB Conversion Activity- Students will have an opportunity to explore the way graphics programs, such as Microsoft Paint, use three numbers from 0 to 255 in order to create 16,777,216 unique colors for pixels on a computer screen with the RGB color system. Discussion points will include the math behind how many more colors could be available for every added bit of memory that may be

assigned to a color in an even more advanced system.**TCC3(9-12)**
Teacher Resources: http://chortle.ccsu.edu/Java5/index.html#07; Java Runtime Environment; Java Development Kit

Unit 2 - Working with Types of Data

Desired Results - Goals, Transfer, Meaning, Acquisition

Established Goals:
2-DA-07 - Represent data using multiple encoding schemes.
2-AP-11 - Create clearly named variables that represent different data types and perform operations on their values.
2-AP-19 - Document programs in order to make them easier to follow, test, and debug.
3A-DA-10 - Evaluate the tradeoffs in how data elements are organized and where data is stored.

Vision of A Learner Attributes: Students will be able to independently use their learning to... ("I can" statements to be demonstrated)
TCC3(9-12): I can integrate relevant information to produce multiple valid solutions.
TCC4(9-12): I can integrate my learning to adapt to experiences in the classroom, career and life.
TI3(9-12): I can formulate and investigate probing questions to further my learning.

Understandings: Students will understand that…

- Information is input to a computing device, abstracted to a binary representation, and then output back in its original form.
- Data is either a primitive data type or an object.
- Values stored in variables are not static. They may change over the course of the running of a computer program.
- Programs are made to interact with a user through the process of input and output.

Essential Questions:

- How is information input to, stored in, and output by a computing device?
- What rules/limitations exist for data storage?
- How do we change stored data?
- How do computer programs interact with a user?

Students will know…

- The capacity limitations for the primitive data types.
- The rules for variable names.
- Computer code utilizes the same order of operations that we learn in mathematics
- The remainder operator performs a division, but outputs the resulting remainder, not quotient.
- Distinct Scanner objects should be created for handling

Students will be able to...

- Write a program that declares variables and assigns values to them.
- Write a program that uses arithmetic operators to change the value stored in a variable.
- Write a program that uses input and output in a meaningful way.

input of different variable types	

Key Vocabulary: Primitive Data Type, Object, Bit, Byte, Short, Long, Int, Double, Float, Char, Moore's Law, Variable, Declaration of a variable, Initialization of a variable, Arithmetic Operator, Remainder Operator (%), Unary Minus/Plus, Input, Output, Scanner Object

Assessment Evidence

Performance Tasks:

Summative Unit Assessment TCC3(9-12), TCC4(9-12), and TI3(9-12) Topics will include:

- Comparing and contrasting the uses of 8 primitive data types.
- Explaining how decimal values are represented in binary and how this leads to round off error.
- Identifying variable names that are, or are not, java-appropriate.
- Determining the output of a simple program using various arithmetic operators.
- Identifying proper uses of input and output in a simple program.

Other Evidence:

Classwork tasks described below

- Money Converter Program
- Input and Output Activity

Online Self-Assessment Quizzes and Review Tasks:

- Primitive Data https://chortle.ccsu.edu/java5/Notes/chap08/chap08quizRev2.html
- Variables and Assignment https://chortle.ccsu.edu/java5/Notes/chap09A/chap09quizRev2.html
- Expressions and Arithmetic Operators https://chortle.ccsu.edu/java5/Notes/chap09B/chap09BquizRev2.html
- Input and Output https://chortle.ccsu.edu/java5/Notes/chap10/fillBlankCh10.html

Learning Plan

In object-oriented programming languages like Java, data is categorized into one of two different categories: primitive data types and objects. Students are asked to consider primitive data types as the basic building blocks (the nuts and bolts, wood, steel, cement, etc.) of a programming language and objects as the things we create with those basic building blocks. In this unit, students learn about these primitive data types, operators that are used with data, and the use of input and output to obtain, modify, and produce results with this data.

- Primitive Data Types Introduction- Students are introduced to the 8 primitive data types of Java: bit, byte, short, long, int, double, float, and char. The data capacity for each will be displayed and students should discuss the relative tradeoff of using a variable type with more capacity versus less. Part of this discussion will involve Moore's law, and how the need to minimize capacity has changed over time. Additionally, students will explore how double type numbers are stored in binary and the possibility for round-off errors when doing

so.**TCC3(9-12)**

- Variables and Assignment Lesson- Students will learn how to declare a variable, rules for creating variable names, and how to assign a value to a variable. A focus will be on the difference between declaring a variable versus initializing it.**TCC3(9-12)**
- Expressions and Arithmetic Operators Activity- Students will write an elementary program that changes values stored in variables through the use of Arithmetic Operators. Although students are already familiar with many of these, as well as the order of operations, the inclusion of the remainder operator(%) and the need to differentiate between the subtraction and unary minus operators will be new information.**TCC3(9-12)**
- Money Converter Program- Students will write a program using the remainder operator to take a hard-coded number of pennies and report out how many dollars, quarters, dimes, nickels, and pennies this would be most efficiently converted to.**TCC4(9-12)**
- Input and Output Activity- For the first time, students will write a program that will be interactive. After learning about how a Scanner object works, students will create a program that given four user-entered values, will find the slope of the line that contains two ordered pairs.**TCC4(9-12)**

Teacher Resources: http://chortle.ccsu.edu/Java5/index.html#07; Java Runtime Environment; Java Development Kit

Unit 3 - Branches and Loops

Desired Results - Goals, Transfer, Meaning, Acquisition

Established Goals:
2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
3B-AP-13 - Illustrate the flow of execution of a recursive algorithm.
3B-AP-21 - Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22 - Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.

Vision of A Learner Attributes: Students will be able to independently use their learning to... ("I can" statements to be demonstrated)
TCC4(9-12): I can integrate my learning to adapt to experiences in the classroom, career and life.
TI3(9-12): I can formulate and investigate probing questions to further my learning.
P4(9-12): I can take-on challenges and continuously engage in my own long-term strategies to overcome them to demonstrate through personal experience that failures are more instructive than discouraging.

Understandings: Students will understand that…
- Control structures, such as conditional statements and loops are abstractions that enable increased complexity to computer programs
- Like tools, different control structures serve different purposes.

Essential Questions:
- How does a computer program use selection and iteration to manage code complexity?
- How does a computer program "make a decision"?
- Why are conditional statements and loops necessary?
- In what ways can loops reduce "code clutter"?

Students will know…
- An if statement can include any number of branched else or else-if statements.
- The behavior of a program depends on the results of conditional statements.
- There are three different types of loops, each with a different structure and purpose.
- A loop allows us to repeat an algorithm a fixed, or variable,

Students will be able to...
- Write programs that incorporate conditional statements in a meaningful way.
- Write programs that incorporate loops in a meaningful way.

number of times.	

Key Vocabulary: If statement, If-else statement, Nested If statement, Loop, While Loop, Counting Loop, Sentinel-controlled Loop, Result-oriented Loop, Nested Loop

Assessment Evidence

Performance Tasks:	Other Evidence:
Summative Unit Assessment TCC4(9-12), TI3(9-12), and P4(9-12) Topics will include: • Correcting a program that does not work in order to make it function as intended. • Identifying the three key components of any loop. • Comparing and contrasting the types of loops, and what they are used for. • Writing a program that calculates the third length of a side of a right triangle. The program will use a conditional statement to determine if the program is finding the length of a leg or hypotenuse and a loop to be used for multiple triangles.	Classwork tasks described below • Check Digit Program • Improved Program with Loop Online Self-Assessment Quizzes and Review Tasks: • If Statements https://chortle.ccsu.edu/java5/Notes/chap12/chap12quizRev2.html • Counting Loops https://chortle.ccsu.edu/java5/Notes/chap16/chap16quizRev2.html • Sentinel-Controlled Loops https://chortle.ccsu.edu/java5/Notes/chap18/fillBlankCh18.html • Result-Controlled Loops https://chortle.ccsu.edu/java5/Notes/chap19/chap19quizRev2.html • Nesting Loops and If statements https://chortle.ccsu.edu/java5/Notes/chap17/fillBlankCh17.html

Learning Plan

In this unit, students will be introduced to the core CS components of selection and iteration. Selection, in terms of computing, refers to a conditional statement that dictates which of two or more possible paths will be followed. The most common structure for this is the "if-then" statement. Iteration is the process of repeating a behavior, often with a change to a condition that dictates the nature and number of repetitions. These two control structures drastically expand a student's ability to create dynamic, robust programs.

- Flowchart Activity- To introduce the concept of conditional statements and loops, students will build flowcharts that model some real world activity. An example can be seen here: https://europe1.discourse-cdn.com/arduino/optimized/4X/c/2/4/c24af5e1f6386d5c03513d747c86a7cebca870f0_2_327x500.png.**TI3(9-12)**
- If statements discussion- Students will be introduced to, and see in the context of code, the various types of conditional statements.

These include: if-then, if-else, and nested if statements. Discussion will be had about which statements are best in different scenarios and why one type would be chosen over another. Emphasis will be placed on focusing on the number of possible outcomes as it relates to the structure of the conditional statements.**TI3(9-12)**

- Check Digit Program- Students will write code that takes an input of an ISBN Number(Semester Course) or a Credit Card Number(AP) and uses the check digit scheme to determine if the inputted number is legitimate. If it is not, the program should state what the correct check digit should be.**TCC4(9-12)**
- Loop discussion- Students will learn about the three types of loops (sentinel-controlled, result-controlled, counting) and the three key components to each (initialized control variable, condition lined to control variable, way to change this variable). Students will discuss when each type is appropriate.**TI3(9-12)**
- Improved Program with Loop- Students will take any previously written code segment and include a loop that, based on user input, repeats the algorithm until the user indicates otherwise.**P4(9-12)**

Teacher Resources: http://chortle.ccsu.edu/Java5/index.html#07; Java Runtime Environment; Java Development Kit

Unit 4 - Advanced Programming Techniques

Desired Results - Goals, Transfer, Meaning, Acquisition

Established Goals:
3B-AP-11 - Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-21 - Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22 - Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.

Vision of A Learner Attributes: Students will be able to independently use their learning to... ("I can" statements to be demonstrated)
TCC3(9-12): I can integrate relevant information to produce multiple valid solutions.
AA1(9-12): I can evaluate different approaches and justify the best pathway to success.
AA4(9-12): I can create opportunities to extend my learning by remaining open-minded in any situation.

Understandings: Students will understand that…
- Common procedures, notations, and structures have been simplified and improved to meet the needs of Java programmers.
- Boolean expressions are only equivalent only if they are equivalent for all combinations of values for the individual variables.
- For loops are typically the most efficient structure of a loop.

Essential Questions:
- How do programming languages change and evolve over time?
- Are there better ways to do what I already know how to do?
- How can we evaluate complex boolean expressions?

Students will know...
- Some shortcut operators perform a function before or after other sections of code are executed.
- Short-circuit evaluation allows programs to only check a later part of a boolean expression if needed, thus negating the possibility of a potential run-time error.
- A truth table allows us to verify the equivalence of boolean expressions.
- According to DeMorgan's Laws, !(A && B) == !A || !B and !(A || B) == !A && !B.

Students will be able to...
- Use Increment, Decrement, and Assignment Operators in program code.
- Complete a truth table.
- Identify equivalent boolean expressions.
- Use for loops in situations that require a loop.

• A for loop puts all three key components of a loop into one line of code.	

Key Vocabulary: Increment Operators, Decrement Operators, Assignment Operators, “not” (!) Operator, “and” (&&) Operator, and “or” (||) Operator, Short-circuit Evaluation, Truth Table, DeMorgan’s Laws, For Statement

Assessment Evidence

Performance Tasks:

Summative Unit Assessment TCC3(9-12), AA1(9-12), and AA4(9-12)
Topics will include:

- Identifying the correct result of using increment, decrement, and assignment operators.
- Completing missing columns in a truth table.
- Matching equivalent boolean expressions based on their truth values.
- Explaining, and demonstrating with an example, the purpose of using short-circuit evaluation
- Rewriting a code segment with a standard loop to contain a for loop, while retaining all functionality.

Other Evidence:

Classwork tasks described below

- Guess Who Activity
- Caesar Cypher Program

Online Self-Assessment Quizzes and Review Tasks:

- Increment, Decrement, and Assignment Operators https://chortle.ccsu.edu/java5/Notes/chap39/chap39quizRev2.html
- Boolean Expressions and Short-circuit Operators https://chortle.ccsu.edu/java5/Notes/chap40/chap40quiz.html
- Truth Tables and DeMorgan’s Laws https://chortle.ccsu.edu/java5/Notes/chap40B/fillBlankCh40B.html
- The For Statement https://chortle.ccsu.edu/java5/Notes/chap42/chap42quizRev2.html

Learning Plan

This unit builds on previous topics in ways that are either covered in more depth, or by introducing upgraded approaches to foundational concepts already learned. Increment, Decrement, and Assignment Operators allow alternate ways to change the values stored in variables. Handling of conditional statements is approached much more deeply through the analysis of truth tables, by the creation of complex boolean statements, and the use of DeMorgan’s Laws to simplify these statements. The For loop takes the three key components of a loop and combines them into a single statement, thus creating code that is easier to troubleshoot and debug.

- Shortcut operator activity- Students will read the following information https://chortle.ccsu.edu/java5/Notes/chap39/ch39_1.html and do a think-pair-share activity on what they have learned.**TCC3(9-12)**

- Truth Table Demonstration- Students will be shown the basic truth tables for a “not” (!), “and” (&&), and “or” (||) operator. Increasingly difficult truth tables will be worked on by students in small groups. Through the use of truth tables, DeMorgan’s Laws will be demonstrated and proven.**AA4(9-12)**
- Guess Who Activity- Students will write a program that, given the least number of pieces of information, stored as boolean variables, will uniquely identify each member of the class.**AA1(9-12)**
- For Loop Discussion- Students will be shown a program that uses a for loop alongside a program that uses a standard loop, but both programs do the exact same thing. Students should highlight the three key components of the standard loop and, with the same color highlighter, indicate where those components show up in the for loop.**TCC3(9-12)**
- Caesar Cypher Program- Students will write a Caesar Cypher program. This program would take in a user input of text as well as a number(x) for the displacement of the letters in the alphabet. The program will use a for loop to iterate through the indexes of the string and change the text by swapping out letters for the ones x spaces over in the alphabet. For example, if x was 3, a->d, b->e, c->f, and so on. The program should then report what the coded message would be.**AA1(9-12)**

Teacher Resources: http://chortle.ccsu.edu/Java5/index.html#07; Java Runtime Environment; Java Development Kit

Unit 5 - Arrays

Desired Results - Goals, Transfer, Meaning, Acquisition

Established Goals:
3A-AP-14 - Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
3B-AP-12 - Compare and contrast fundamental data structures and their uses.
3B-AP-21 - Develop and use a series of test cases to verify that a program performs according to its design specifications.
3B-AP-22 - Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
3B-AP-23 - Evaluate key qualities of a program through a process such as a code review.

Vision of A Learner Attributes: Students will be able to independently use their learning to... ("I can" statements to be demonstrated)

CCE2(9-12): I can give and receive actionable and relevant feedback with openness to be able to determine meaningful revisions for success.
TI3(9-12): I can formulate and investigate probing questions to further my learning.
P4(9-12): I can take-on challenges and continuously engage in my own long-term strategies to overcome them to demonstrate through personal experience that failures are more instructive than discouraging.

Understandings: Students will understand that…

- An Array is a data collection tool that allows for organization of real world lists such as rosters, statistics, testing results, etc. using indexes and a series of common searching and sorting algorithms.
- Variables can be referenced through the name of a collection object and the index of that object more easily than through the use of numerous variable names.

Essential Questions:

- How do arrays, and other collection objects, manage code complexity?
- When is it appropriate to use an array?

Students will know...

- Each cell of a numeric array is initialized to zero; each cell of an array of references is initialized to null.

Students will be able to...

- Initialize, adjust, and read values from an array in a variety of Java programs.
- Construct two dimensional arrays for use in Java programs.

• A statement of the form arrayB=arrayA will only copy the object reference in arrayA to arrayB, it will not change the contents of arrays. • Because an array index starts at 0, but counting loops usually start at 1, there is a high likelihood of error without careful attention to detail. • Arrays can contain string references inside cells. • A 2-d array is a 1-d array of 1-d arrays.	
Key Vocabulary: Array, Index, 1-d Array, 2-d Array, Row-Column Order	
Assessment Evidence	
Performance Tasks: **Summative Unit Assessment CCE2(9-12), TI3(9-12), and P4(9-12)** Topics will include: • Explaining how and when an array should be used. • Determine if notation used to interact with an array is syntax and error free. • Complete a program that finds the average value of the integers stored in an array. • Generate a 2-d array of specific size such that each cell is the product of its indices	**Other Evidence:** Classwork tasks described below • Grade List Program • Seating Chart Activity Online Self-Assessment Quizzes and Review Tasks: • Introduction to Arrays https://chortle.ccsu.edu/java5/Notes/chap46/chap46quizRev2.html • Common Array Algorithms https://chortle.ccsu.edu/java5/Notes/chap47/chap47quizRev2.html • Arrays as Parameters https://chortle.ccsu.edu/java5/Notes/chap48/fillBlankCh48.html • Methods that Change Arrays https://chortle.ccsu.edu/java5/Notes/chap49A/chap49AquizRev2.html • Two-dimensional Arrays https://chortle.ccsu.edu/java5/Notes/chap49C/chap49CquizRev2.html
Learning Plan	

In this unit, students learn the data structure of an array. Arrays allow for the storage of multiple variables without the need for individual variables. In some cases, an array is not only more efficient, but the only way for a program to process a collection of some number of similar variables.

- Array introduction- Students will be shown the basics for creating an array, storing, accessing, and changing values in the array, and an algorithm for iterating through the array with a loop in order to perform some function on the array such as printing out the contents. Students will write their first array program, one that creates a list of random numbers, in groups of 2.**CCE2(9-12)**
- Grade list program- Students will write a program that stores grades in a 1-d array. Scores can be accessed, modified, or deleted based on input from the user. As an extension, students may try to write an algorithm to figure out the average of the grades at any point.**P4(9-12)**
- 2-d array lesson- The concept of a 2-d array is a difficult one for students because Java processes a 2-d array as a 1-d array of 1-d arrays. The visualization of this for students takes some time, but is best explained with the real world metaphor of a series of hallways that each contain a series of lockers.**TI3(9-12)**
- Seating chart activity- Students will write a program that makes a seating chart as a 2-d array. User input allows strings of names to be input into various spots in the chart and then the resulting seating chart is printed out.**P4(9-12)**

Teacher Resources: http://chortle.ccsu.edu/Java5/index.html#07; Java Runtime Environment; Java Development Kit