

Madison Public Schools

AP Computer Science A Curriculum

Written by:

Jason Erdreich

Reviewed by:

Daniel Ross

Asst. Superintendent of Curriculum, Instruction, and Personnel

Approval date:

August 20, 2019

Members of the Board of Education:

Leslie Lajewski, President

Heather Reddy, Vice President

Sarah Fischer

Johanna Habib

David Irwin

Thomas Piskula

Abi Singh

Pam Yousey

Madison Public Schools
359 Woodland Road
Madison, NJ 07940
www.madisonpublicschools.org

Course Overview

Description

This year-long AP course embraces problem solving, hardware, algorithms and perspectives that help people utilize computers to solve real-world problems in everyday life. The AP Computer Science A course introduces students to computer science with fundamental topics that include problem solving, design strategies and methodologies, organization of data (data structures), approaches to processing data (algorithms), analysis of potential solutions, and the ethical and social implications of computing. The course emphasizes both object-oriented and imperative problem solving and design. These techniques represent proven approaches for development solutions that can scale up from small, simple problems to large, complex problems. Students will be prepared to take the AP Computer Science A exam and the end of the course.

Goals

Students will be able to...

- Design and implement computer-based solutions to problems.
- Use and implement commonly used algorithms and data structures.
- Develop and select appropriate algorithms and data structures to solve new problems.
- Write solutions fluently an object-oriented paradigm
- Write, run, test and debug solutions in the Java programming language
- Read and understand programs consisting of several classes and interacting objects
- Read and understand a description of the design and development process
- Understand the ethical and social implications of computer use.

Materials

Core:

- AP CS A Curricula and Modules, Edhesive
- Introduction to Programming using Java, David J. Eck
- Rep.it Programming IDE
- DrJava Programming IDE

Supplemental:

- Code Runner
- Khan Academy
- Code.org
- CompuScholar

Resources

[Suggested activities and resources page](#)

Benchmark Assessments

Benchmark assessments are at the end of each module and focus on the main ideas and anchor standards of the course.

Modifications and Adaptations for Special Needs Learners

(Gifted and Talented Students, English Language Learners, Students with Special Needs, At-Risk Students, and Students with 504 Plans)

Scope and Sequence (Pacing Guide)

Unit Number	Topic of Study	Duration (Weeks Taught)
1	Introduction to Java	4
2	Conditionals and Loops	3
3	Strings and One-Dimensional Arrays	5
4	Methods	4
5	User-Defined Classes	4
6	Advanced Classes	5
7	Algorithms - Searching and Sorting	3
8	Two-Dimensional Arrays	2
9	AP Exam Review	3

Unit 1 Overview
Unit Title: Introduction to Java
Unit Summary: In this introduction unit, students will learn the basics of programming in Java. The content of this unit emphasises on correct programming style and formatting as students experiment with creating their first programs. Students will be introduced to the resources they have access to for the duration of the course in order to assist with learning the content. This unit is intended to set a foundation of vocabulary and understanding for the following units.
Suggested Pacing: 10 lessons
Learning Targets
Unit Essential Questions: <ul style="list-style-type: none"> • What is code? What can we do with code? • How is code used to solve everyday problems in our lives? • How can we write code to interact with and control computers?
Unit Enduring Understandings: <ul style="list-style-type: none"> • Basic Input and Output • Variables • Number Operations including +, -, *, /, % and the AMth.java functions: static int abs(int x), static double abs(double x), static double pow(double base, double exponent), static double sqrt(double x), static double random() // returns a double in the range [0.0, 1.0) • Data Types • Comments and Program Design • Error Correction
Evidence of Learning
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets
Summative Assessments: Lesson quizzes 1.1-1.9, Unit Quiz 1.1, Exam 1.1
Alternative Assessments: Assignment 1.1: Movie Ratings, Assignment 1.2: Timer Converter, Lesson mini coding activities, AP-style multiple choice practice

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLS)
<p>Write their first lines of code in Java</p> <p>Use the <code>system.out.print</code> and <code>system.out.println</code> commands to print text and understand the difference between these commands</p> <p>Encounter the String data type, and know that strings are specified using double quotes</p>	<p>Crucial vocabulary for working with Javascript</p> <p>How to navigate through the programming IDE</p> <p>How to use the <code>system.out.print</code> function in order to display messages and status from code</p> <p>Vocabulary: Code, java, IDE, error, debug, print, input, output,</p>	<p>Create a "Hello World" program using the Java programming IDE</p> <p>Lesson 1.1 coding activities through Edhesive</p> <p>Lesson 1.1 quiz through Edhesive</p>	<p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p>
<p>Understand the purpose of escape characters in strings and write programs using several different escape characters</p> <p>Understand the use and importance of comments in code</p>	<p>How to write strings in a program using a Java coding language</p> <p>When comments should be added to programs and how to add them</p> <p>Vocabulary: String, comment, block, monospace</p>	<p>Escape the Characters assignment</p> <p>Lesson 1.2 coding activities through Edhesive</p> <p>Lesson 1.2 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p>
<p>Write programs which use user input</p> <p>Gain some understanding of what happens when a variable is declared</p> <p>Use string concatenation to create strings from multiple parts</p>	<p>How to declare a variable and use it to store values</p> <p>When variables would be appropriately used</p> <p>Vocabulary: Declaration, variables, scanner, concatenation, print, inputs</p>	<p>User input and variables assignment</p> <p>Lesson 1.3 coding activities through Edhesive</p> <p>Lesson 1.3 quiz through Edhesive</p>	<p>3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information.</p> <p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>A-SSE.1a Interpret expressions that represent a quantity in terms of its context.</p>
<p>Write programs with variables of the int, double and char data types</p> <p>Understand the importance of matching inputs to the correct data type</p>	<p>Different data types in terms of variable formats</p> <p>How to determine which type of variable should be used</p> <p>Vocabulary: Int, double, char, data type, ASCII, Unicode, runtime error</p>	<p>Data types assignment</p> <p>Lesson 1.4 coding activities through Edhesive</p> <p>Lesson 1.4 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p>
<p>Perform calculations with int and double values</p> <p>Understand how integer division works in Java, and how to get a more precise value for a division if desired</p>	<p>How to complete arithmetic calculations in javascript coding applications</p> <p>Applications for algebraic equations from math concepts in a programming language's functionality</p> <p>Vocabulary: arithmetic calculations, operators, integers, unary. Variables, data types</p>	<p>Number calculations assignment</p> <p>Lesson 1.5 coding activities through Edhesive</p> <p>Lesson 1.5 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.</p> <p>A-SSE.2 Use the structure of an expression to identify ways to rewrite it.</p>
<p>Perform casts between int and double data types in both directions</p>	<p>When data conversions will benefit the implementation of a program</p>	<p>Numeric casts assignment</p> <p>Lesson 1.6 coding activities through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p>

<p>Understand the difference between a widening conversion and a narrowing conversion</p>	<p>How to use multiple forms of data conversions in a javascript programming language</p> <p>Vocabulary: Conversions, widening conversion, narrowing conversion, casts, truncate</p>	<p>Lesson 1.6 quiz through Edhesive</p> <p>Assignment 1.1: Movie Ratings</p>	<p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>
<p>Understand the definition of modular division and be able to manually calculate the value of modular expressions (e.g. 7%3)</p> <p>Use modular division in Java to solve problems</p>	<p>How to complete modular division calculations in order to determine remainders</p> <p>What previously learned algebraic skills can be applied in coding arithmetic</p> <p>Vocabulary: Modular Division, remainder, operator, change, percent function</p>	<p>Modular division assignment</p> <p>Lesson 1.7 coding activities through Edhesive</p> <p>Lesson 1.7 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>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).</p>
<p>Identify what the term concatenation means</p> <p>Understand that the + operator causes strings to be concatenated and integers to be added, and that it is important therefore to know what type of data Java thinks it is working with</p>	<p>How to join two strings together through concatenation</p> <p>How Java allocates memory when working with strings</p> <p>What order of operations can be utilized to avoid interpreting strings as numbers</p> <p>Vocabulary: Concatenation, strings, outputs, libraries, APIs, order of operations</p>	<p>String and number output assignment</p> <p>Lesson 1.8 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Use commands from the Math class in Java</p> <p>Understand that extra functionality can be added to programs by importing new commands</p>	<p>How to access math methods through importing libraries</p> <p>Vocabulary: Math class, import, libraries, package, function</p>	<p>Math functions assignment</p> <p>Lesson 1.9 coding activities through Edhesive</p> <p>Lesson 1.9 quiz through Edhesive</p> <p>Unit 1.1 quiz</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>A-SSE.1a Interpret expressions that represent a quantity in terms of its context.</p>
<p>Identify what the term roundoff error means and when/why it can occur</p> <p>Explore a strategy to avoid round-off errors in calculations, and write code using this strategy</p>	<p>The difference between Java storing doubles and ints</p> <p>When approximations are appropriate and necessary in coding applications</p> <p>Vocabulary: Roundoff error, round, cast, calculations, double, int</p>	<p>Roundoff Error assignment</p> <p>Lesson 1.10 coding activities through Edhesive</p> <p>Lesson 1.10 quiz through Edhesive</p> <p>Unit 1.1 Exam</p> <p>Assignment 1.2: Time Converter</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>

Unit 2 Overview
Unit Title: Conditionals and Loops
Unit Summary: In this unit, students will continue to develop their skills in computer programming through the utilization of the Javascript programming language. Students will be introduced to some of the basic constructs of programming, including conditionals and loops. In addition, students are introduced to debugging techniques and tracing code. In this unit, students will have the opportunity to apply their skills and understandings through solving real-world problems using intermediate coding techniques.
Suggested Pacing: 10 lessons
Learning Targets
Unit Essential Questions: <ul style="list-style-type: none"> • How can programming be optimized? • What constructs exist that can increase efficiency in our programs? • As code becomes more complex, what strategies can we use to troubleshoot problems that arise?
Unit Enduring Understandings: <ul style="list-style-type: none"> • If Statements • Boolean Operations • DeMorgan's Law • Short Circuit Evaluation • While Loops • Java constants Integer.MIN_VALUE and Integer.MAX_VALUE
Evidence of Learning
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets
Summative Assessments: Lesson quizzes 1.11-1.19, Unit Quiz 1.2, Exam 1.2
Alternative Assessments: Assignment 1.3: Crack the Code!, Lesson mini coding activities, AP-style multiple choice practice

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLs)
<p>Understand that programs can have commands which only run if certain conditions are true (e.g. based on the value of an inputted variable)</p> <p>Write code using if statements in Java</p>	<p>Appropriate syntax for creating if statements within their programs</p> <p>How to utilize if statements as gates to enable autonomy in programming applications</p> <p>Vocabulary: Conditions, if statements, gates, comparisons</p>	<p>Simple ifs assignment</p> <p>Lesson 1.11 coding activities through Edhesive</p> <p>Lesson 1.11 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p>
<p>Understand the meaning of the operators ==, <, >, <=, >= and != in Java, and use these to create boolean statements comparing values</p>	<p>How to select the appropriate operator when creating boolean statements</p> <p>How to use if statements to create comparisons between multiple values and variable types</p> <p>Vocabulary: Operators, boolean statements, edge cases, boundary cases</p>	<p>Making decisions assignment</p> <p>Lesson 1.12 coding activities through Edhesive</p> <p>Lesson 1.12 quiz through Edhesive</p>	<p>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).</p> <p>A-SSE.1b Interpret expressions that represent a quantity in terms of its context.</p>
<p>Write programs which use else and else if statements</p> <p>Understand how Java interprets a structure with multiple conditional statements</p>	<p>Appropriate syntax to create else and else if statements following if functions</p> <p>How to create longer blocks through the combination of boolean statements</p> <p>Vocabulary: Else statements, else if statements, TRUE, FALSE</p>	<p>Else assignment</p> <p>Lesson 1.13 coding activities through Edhesive</p> <p>Lesson 1.13 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p>
<p>Understand the meaning of the boolean operators &&, , and !</p> <p>Write programs which use boolean operators to create compound boolean statements</p> <p>Use truth tables to determine when boolean expressions are true or false</p>	<p>How to read and create truth tables</p> <p>When complex boolean expressions should be used and how to create them</p> <p>Vocabulary: Boolean operators, truth tables, OR, NOT</p>	<p>Booleans and Truth Tables assignment</p> <p>Lesson 1.14 coding activities through Edhesive</p> <p>Lesson 1.14 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>F-IF.9 Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions).</p>
<p>Identify how to use short-circuit evaluation to save time when evaluating boolean statements</p> <p>Understand that Java uses short-circuit evaluation when evaluating boolean statements and how to take advantage of this</p>	<p>How to choose boolean statements in order to avoid coding errors</p> <p>The functionality of short-circuit evaluation</p> <p>Vocabulary: Short-circuit evaluation, compound boolean statements,</p>	<p>Short-circuit evaluation assignment</p> <p>Lesson 1.15 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p>
<p>Use De Morgan's law to write equivalent boolean statements</p> <p>Identify the inverses of the statements $x = y$, $x < y$ and $x > y$</p>	<p>How to use De Morgan's law in creating boolean statements that are equivalent</p> <p>Vocabulary: De Morgan's Law, Inverses</p>	<p>De Morgan's Law assignment</p> <p>Lesson 1.16 quiz through Edhesive</p> <p>Unit 1.2 quiz</p>	<p>A-SSE.3c Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.</p>

<p>Write code using while loops to repeat blocks of commands</p> <p>Understand that a while loop repeats as long as a boolean condition is true</p> <p>Identify what is meant by the scope of a variable</p>	<p>The concept of looping in coding applications</p> <p>How to create autonomous repetition through the implementation of a while loop</p> <p>Vocabulary: Loops, while command, variable scope, repetition blocks</p>	<p>While Loops assignment</p> <p>Lesson 1.17 coding activities through Edhesive</p> <p>Lesson 1.17 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Understand how to read and interpret code so that the results of running it can be determined</p>	<p>Skills and techniques needed to efficiently and accurately trace code</p> <p>How and why a trace table is created</p> <p>How to associate variables with outputs to determine program functionality through fundamental reasoning</p> <p>Vocabulary: Code tracing, trace table troubleshooting, debugging, variable association</p>	<p>Tracing Code assignment</p> <p>Lesson 1.18 quiz through Edhesive</p>	<p>3B-AP-21 Develop and use a series of test cases to verify that a program performs according to its design specifications.</p>
<p>Practice tracing a while loop which performs a more complex function</p>	<p>How to apply previously learned tracing skills on while loop functions</p> <p>Vocabulary: Mod operator, tracing code, while loops</p>	<p>More loops assignment</p> <p>Lesson 1.19 quiz through Edhesive</p>	<p>8.2.2.E.4 Debug an algorithm (i.e., correct an error).</p>
<p>Understand the purpose of flag variables in code and write code which uses flag variables</p> <p>Write programs which find the maximum and minimum of values entered</p>	<p>How to use the Java constants Integer.MIN_VALUE and Integer.MAX_VALUE</p> <p>When to use flag variables function and how to implement them in a program</p> <p>Vocabulary: Flag variable, maximum value, minimum value, constants</p>	<p>Flag Variables assignment</p> <p>Lesson 1.20 coding activities through Edhesive</p> <p>Lesson 1.20 quiz through Edhesive</p> <p>Unit 1.2 exam</p> <p>Assignment 1.3: Crack the Code!</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>3B-AP-21 Develop and use a series of test cases to verify that a program performs according to its design specifications.</p> <p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>

Unit 3 Overview
Unit Title: Strings and One-Dimensional Arrays
Unit Summary: In this unit, students will begin to transition from building coding techniques to working with advanced coding methods. Starting with designing classes and algorithms, students will develop advanced objects and data types using strings and arrays. Students will learn to create standard algorithms on data structures such as finding max, min, transversals, insertions, and deletions through coding arrays.
Suggested Pacing: 11 lessons
Learning Targets
Unit Essential Questions: <ul style="list-style-type: none"> • How can complex programs be optimized to run autonomously? • How can large data sets be created and analyzed rapidly through the use of algorithms and arrays? • What functions exist to analyze data sets for a real-world application in math and science? • What framework exists to organize the structure of complex programs? Why is this framework important?
Unit Enduring Understandings: <ul style="list-style-type: none"> • Constructors • Concatenation • For Loops • String methods: equals (), charAt(), toUpperCase(), substring() • One-Dimensional Arrays • Transversals, Insertion and Deletion with Arrays • Arrays of Objects
Evidence of Learning
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets
Summative Assessments: Lesson quizzes 1.21-1.30, Unit Quiz 1.3, Exam 1.3
Alternative Assessments: Assignment 1.4: String Shortener, Assignment 1.5: Array Statistics, Assignment 1.6: Merge Arrays. Real-world Application: Algorithms and Arrays in Chatbots, Lesson mini coding activities, AP-style multiple choice practice

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLs)
<p>Identify the difference between primitive and class data types, and how Java stores these in computer memory</p> <p>Understand the effect that this has when comparing class data vs comparing primitive data</p>	<p>How to create class functions in Java programming</p> <p>How to access data allocated through primitive and class memory systems</p> <p>Vocabulary: Strings, memory, class data, primitive data,</p>	<p>Class allocation assignment</p> <p>Lesson 1.21 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p>
<p>Understand that a string is made up of characters which can be stored with indices starting at 0 and increasing sequentially</p> <p>Identify the purpose and behavior of several different functions from the String class and use these in code</p>	<p>How to develop a method to compare strings of data</p> <p>The differences between comparative functions and when to use them to analyze data types</p> <p>Vocabulary: Strings, class, character, call, method, comparisons</p>	<p>Lesson 1.22 coding activities through Edhesive</p> <p>Lesson 1.22 quiz through Edhesive</p> <p>Assignment 1.4: String Shortener</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>F-IF.9 Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>
<p>Identify what an array data type is, and how to declare and initialize an array</p> <p>Identify how to extract and edit data which is contained in an array</p> <p>Understand how array elements are indexed</p>	<p>What an array function is and how to code it</p> <p>How to access and utilize large amounts of data through the use of arrays</p> <p>Vocabulary: Arrays, initializer list</p>	<p>Coding arrays assignment</p> <p>Lesson 1.23 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Understand the syntax of a for loop in Java</p> <p>Write code which uses for loops to repeat commands, including code which loops through the values in an array</p>	<p>How to create a for loop using proper syntax</p> <p>What repetition and autonomy can be achieved with for loops</p> <p>Vocabulary: For Loop, array, i, list</p>	<p>For Loops assignment</p> <p>Lesson 1.24 coding activities through Edhesive</p> <p>Lesson 1.24 quiz through Edhesive</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p> <p>CRP11 Use technology to enhance productivity.</p>
<p>Understand the meaning of the term “algorithm” and the features which make a good algorithm</p> <p>Explore a simple mathematical algorithm and see how this can be translated into Java code</p>	<p>What are the elements of a good algorithm</p> <p>How to create and program an algorithm in Java</p> <p>How to interpret the sieve of eratosthenes function</p> <p>Vocabulary: Algorithm, sieve of eratosthenes</p>	<p>Algorithms assignment</p> <p>Lesson 1.25 quiz through Edhesive</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>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).</p>
<p>Identify how algorithms can be used on arrays and how these can be coded in Java</p>	<p>How to solve complex problems using arrays by integrating algorithms</p>	<p>Algorithms on Arrays assignment</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p>

	Vocabulary: Array algorithms, insertion, deletion	Lesson 1.26 quiz through Edhesive	8.2.2.E.4 Debug an algorithm (i.e., correct an error).
Understand how a linear search algorithm finds values in an array, and see an implementation of this algorithm in Java	How linear programming impacts all elements of a function How to perform searches using algorithms in Java programming Vocabulary: Linear search, algorithm, array, for loop	Searching algorithms assignment Unit 1.3 Quiz Assignment 1.5: Array Statistics	3B-AP-10 Use and adapt classic algorithms to solve computational problems. 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). 9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.
Understand that arrays can contain class data as well as primitive data Analyze examples of code which uses arrays containing the String data type	How to optimize data types using arrays Vocabulary: Strings, arrays, loops, null	Arrays of Strings assignment Lesson 1.28 quiz through Edhesive	3B-AP-12 Compare and contrast fundamental data structures and their uses.
Write code which accesses elements of a String array and applies String methods to these elements	How to access methods through an array Simplify and optimize code through the changing of syntax in calling a string array Vocabulary: Methods, string array	Accessing Methods assignment Lesson 1.29 coding activities through Edhesive Lesson 1.29 quiz through Edhesive	3B-AP-12 Compare and contrast fundamental data structures and their uses. A-SSE.1b Interpret expressions that represent a quantity in terms of its context.
Create a program by writing code in Java by hand on paper Understand the general ideas behind how free response code is marked on the AP exam	Strategies for creating a free response program Techniques for writing code by hand Vocabulary: Strings, algorithm, array, loop, error	Processing Arrays of Strings assignment Lesson 1.30 coding activities through Edhesive Lesson 1.30 quiz through Edhesive	3B-AP-23 Evaluate key qualities of a program through a process such as a code review.
Understand how numbers can be represented in binary and octal number systems Identify how to convert numbers between binary, octal and decimal representations	How to make calculations and conversions using the binary number system How to make calculations and conversions using the octal number system Vocabulary: Binary system, octal system, decimal system	Binary & Octal assignment Lesson 1.30.5 coding activities through Edhesive Lesson 1.30.5 quiz through Edhesive Assignment 1.6: Merge Arrays Real-world Application: Algorithms and Arrays in Chatbots Unit 1.3 exam	3B-AP-12 Compare and contrast fundamental data structures and their uses. 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). 9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.

Unit 4 Overview	
Unit Title: Methods	
Unit Summary: In this unit, students continue to develop advanced programming skills as they begin to design their own custom functions. Students will be introduced to the complex concept of methods, as well as the different ways to code and utilize methods in their programs. Students will have the opportunity to design and test their own custom methods that can be used to solve problems through code. Recursion will also be introduced and direct comparisons between iteration and recursion are made within this unit as well.	
Suggested Pacing: 7 lessons	
Learning Targets	
Unit Essential Questions: <ul style="list-style-type: none"> • How are programs unique? How are they similar? • What do methods achieve when integrating them into a program? • What are the various types of methods and when should they each be utilized? • What techniques must be used to develop custom programming method functions? 	
Unit Enduring Understandings: <ul style="list-style-type: none"> • Void Methods • Methods that Return a Single Value • Overloading Methods • Parameter Passing • Primitive vs. Object Parameters • Recursion 	
Evidence of Learning	
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets	
Summative Assessments: Lesson quizzes 1.31-1.37, Unit Quiz 1.4, Exam 1.4	
Alternative Assessments: Assignment 1.7: Methods Sampler Platter, Lesson mini coding activities, AP-style multiple choice practice	

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSL)
<p>Define the definition of a method in Java</p> <p>Understand why methods can be a useful tool in programming</p> <p>Identify the syntax for defining and calling a void method in Java</p>	<p>What methods do in a Java programming application</p> <p>The differences between a void method and public method, as well as how and when to use each</p> <p>Vocabulary: Method, main, call, void methods, public methods</p>	<p>Void Methods assignment</p> <p>Lesson 1.31 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Define parameter in Javascript programming applications</p> <p>Create void methods in Java which use parameters</p>	<p>The correct syntax of a method function</p> <p>How to create a custom void method using parameters</p> <p>Vocabulary: Parameter, void method, header, argument.</p>	<p>Parameters assignment</p> <p>Lesson 1.32 coding activities through Edhesive</p> <p>Lesson 1.32 quiz through Edhesive</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p>
<p>Understand the difference in the way java treats class type and primitive type parameters</p> <p>Predict the values of variables after methods are called with those variables passed as parameters</p>	<p>Changing a method parameter after the method has completed has no effect</p> <p>Changing a class parameter will carry throughout regardless of order</p> <p>Vocabulary: Primitive parameters, class parameters, method, array</p>	<p>Primitive vs. Class assignment</p> <p>Lesson 1.33 coding activities through Edhesive</p> <p>Lesson 1.33 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Understand that methods can return values when called</p> <p>Create methods which return values and write code which catches values returned from methods which are called</p>	<p>How to return variable values through the creation of a custom method in Java</p> <p>Vocabulary: Return Statements</p>	<p>Return Methods assignment</p> <p>Lesson 1.34 coding activities through Edhesive</p> <p>Lesson 1.34 quiz through Edhesive</p> <p>Unit 1.4 Quiz</p> <p>Assignment 1.7: Methods Sampler Platter</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.</p> <p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>
<p>Understand the meaning of overloading in Java</p> <p>Create overloaded methods which have the same name, but take different parameters and may behave differently</p>	<p>Overloading is a useful technique to make programming easier</p> <p>Vocabulary: Overloading, polymorphism, uniform interface</p>	<p>Overloaded Methods assignment</p> <p>Lesson 1.35 coding activities through Edhesive</p> <p>Lesson 1.35 quiz through Edhesive</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>3B-AP-23 Evaluate key qualities of a program through a process such as a code review.</p>

<p>Understand that a method may call itself and that this is called recursion</p> <p>Trace code which uses recursion and predict its results</p>	<p>How to queue methods through the use of recursion</p> <p>What the last in first out method means in Java programming</p> <p>How to reverse the calling order of methods by the recursion process</p> <p>Vocabulary: Recursion, tracing, last in first out</p>	<p>Intro to Recursion assignment</p> <p>Lesson 1.36 quiz through Edhesive</p>	<p>3B-AP-14</p> <p>Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p>
<p>Understand recursive methods which return values</p> <p>Develop trace functions using recursive methods</p>	<p>How to apply previously learned skills of return methods with recursion techniques</p> <p>Vocabulary: Recursion, tracing, returns</p>	<p>Recursive Functions With Returns assignment</p> <p>Lesson 1.37 quiz through Edhesive</p> <p>Unit 1.4 Exam</p>	<p>3B-AP-14</p> <p>Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>A-SSE.1b</p> <p>Interpret expressions that represent a quantity in terms of its context.</p>

Unit 5 Overview	
Unit Title: User-Defined Classes	
Unit Summary: In this unit, students begin their second term of the AP Computer Science A course. Students transition from using classes and objects to designing and coding their own classes. This unit is also structured to serve as a review of the topics from Unit 1 as all of the programming techniques used in Term 1. In particular, basic arrays and ArrayLists are compared and contrasted. This unit also introduces the first of the AP Labs: Magpie that facilitates a discussion and reflection of the social, cultural and ethical implications of chatbots and artificial intelligence.	
Suggested Pacing: 6 lessons	
Learning Targets	
Unit Essential Questions: <ul style="list-style-type: none"> • What are the limitations when creating custom classes? What are the benefits that custom classes can provide to a program? • How can we program functions to solve mathematical and scientific expressions? • What social, cultural, and ethical concerns exist when developing mainstreamed programs? 	
Unit Enduring Understandings: <ul style="list-style-type: none"> • ArrayList Class • java.util.List interface • Constructors • Encapsulation via private variables and accessor and mutator methods • Private/Public • Writing equals() toString() methods • Static Variables and Methods • Wrapper Classes 	
Evidence of Learning	
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets	
Summative Assessments: Lesson quizzes 2.1-2.6, Unit Quiz 2.1, Exam 2.1	
Alternative Assessments: Assignment 2.1: Fraction, Assignment 2.2: Boxcar & Freight Train, AP Lab, Lesson mini coding activities, AP-style multiple choice practice	

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLs)
<p>Understand that an ArrayList is used to hold multiple data as with an array, but that it can only hold class type data and not primitives</p> <p>Write code which declares and adds data to an ArrayList</p> <p>Identify some of the methods which can be called by objects of the ArrayList class</p>	<p>The different ArrayList methods built into Java programming languages</p> <p>How to create an ArrayList using previously learning skills and abilities</p> <p>Vocabulary: ArrayList, Scanner Class, list.add(e), list.add(i, e), list.set(i, e), list.get(i), list.remove(i)</p>	<p>ArrayList assignment</p> <p>Lesson 2.1 coding activities through Edhesive</p> <p>Lesson 2.1 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p>
<p>Write code which implements a for-each loop in Java</p> <p>Understand the advantages and limitations of using a for-each loop versus a standard for loop</p>	<p>The differences between a primary for loop and the for-each loop functions</p> <p>The proper syntax for creating a for-each loop in Java programming</p> <p>Vocabulary: For-each loop, ArrayList</p>	<p>The For-Each Loop assignment</p> <p>Lesson 2.2 coding activities through Edhesive</p> <p>Lesson 2.2 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p> <p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p>
<p>Determine how to properly define a new public class in Java and the conventions in doing this</p> <p>Understand that classes require a constructor so that objects of that class can be created</p> <p>Identify how to declare variables, create constructors and methods for a new class</p>	<p>The three components needed to declare class function</p> <p>How to create custom class functions using principle techniques</p> <p>Vocabulary: Class, constructor, method, Accessors, Mutators, ArrayList</p>	<p>The basics of Classes assignment</p> <p>Lesson 2.3 quiz through Edhesive</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p>
<p>Determine how to call other methods and constructors from within a class and understand the benefit of doing this</p> <p>Identify an example of how multiple constructors may be implemented in a class</p>	<p>A class can have multiple constructors</p> <p>How to use the overloading method in order to create multiple constructors easily</p> <p>Vocabulary: Constructors, overloading, class, method</p>	<p>Constructors assignment</p> <p>Lesson 2.4 coding activities through Edhesive</p> <p>Lesson 2.4 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Understand the difference between static and instance variables and methods</p> <p>Determine some uses of static variables and methods</p> <p>Understand the modifier final prevents the value of a variable being changed</p>	<p>How to share variables through the use of static variable types</p> <p>When to use static variables in comparison to instance variables in a program</p> <p>Vocabulary: Static variables, instance variables, method, modifier</p>	<p>Static Vs. Instance assignment</p> <p>Lesson 2.5 coding activities through Edhesive</p> <p>Lesson 2.5 quiz through Edhesive</p> <p>AP Labs Introduction</p> <p>AP Labs 1: Magpie</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>F-IF.9 Compare properties of two functions each represented in a different way (algebraically, graphically, numerically in tables, or by verbal descriptions).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>

<p>Identify strategies for answering questions on the AP Computer Science A exam</p> <p>Investigate worked examples of both multiple choice and free response AP style questions</p>	<p>The AP exam is a written paper</p> <p>Techniques to work with code without the use of a computer</p> <p>Vocabulary: Free responses, edge cases, trace code</p>	<p>AP Exam Review assignment</p> <p>AP Review questions</p> <p>Lesson 2.6 quiz through Edhesive</p> <p>Unit 2.1 Exam</p> <p>Assignment 2.2 Part 1: Boxcar</p> <p>Assignment 2.2 Part 2: Freight Train</p>	<p>3B-AP-23 Evaluate key qualities of a program through a process such as a code review.</p> <p>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).</p> <p>CRP5 Consider the environmental, social and economic impacts of decisions</p>
--	---	---	--

Unit 6 Overview
Unit Title: Advanced Classes
Unit Summary: In this unit students continue to work with class design by covering the more advanced topics including polymorphism, inheritance, abstraction and interfaces. By learning to these advanced techniques, students will be able to create complex programs that utilize many of the concepts they have learned in previous lessons to solve real-world problems. Students will continue working with the AP Labs in order to use code to solve real-world challenges.
Suggested Pacing: 6 lessons
Learning Targets
Unit Essential Questions: <ul style="list-style-type: none"> • What complexity can be added to classes in order to provide more functionality in our code? • How can we apply previously learned content to simplify our coding functions? • What additional everyday problems are solved through the use of code?
Unit Enduring Understandings: <ul style="list-style-type: none"> • Inheritance • This vs. Super • Abstract Classes • Interfaces including the List and Comparable Interface
Evidence of Learning
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets
Summative Assessments: Lesson quizzes 2.7-2.12, Unit Quiz 2.2, Exam 2.2
Alternative Assessments: Assignment 2.3: Ultimate Frisbee, Assignment 2.4: Fraction Comparable, AP Lab, Lesson mini coding activities, AP-style multiple choice practice

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSL)
<p>Determine how to create a subclass of an existing class</p> <p>Understand that public methods are inherited by the subclass of a class and can be accessed from within the class by using the keyword super</p>	<p>How to use the inheritance technique to create a subclass</p> <p>The differences between the different forms of classes as well as the proper syntax for coding each format</p> <p>Vocabulary: Inheritance, subclass, parent class, child class, super class extends</p>	<p>Inheritance assignment</p> <p>Lesson 2.7 quiz through Edhesive</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</p>
<p>Understand that methods inherited by a subclass can be overwritten in the definition for that subclass</p> <p>Identify an example of a child class in which methods from the parent class are overwritten</p>	<p>Methods in the subclass override counterparts in the superclass</p> <p>The differences between overriding and overloading</p> <p>Vocabulary: Override, subclass, superclass</p>	<p>Inheritance Overriding Methods assignment</p> <p>Lesson 2.8 quiz through Edhesive</p> <p>Assignment 2.3: Ultimate Frisbee</p> <p>Unit 2.2 Quiz</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>
<p>Understand the properties of an abstract class and learn how to extend an abstract class with a concrete class that can be instantiated.</p> <p>Identify an example of an abstract class and a non-abstract subclass which extends it properly</p>	<p>Java will assume that all child classes without those coded methods are abstract until all abstract methods are coded by child classes</p> <p>Vocabulary: Abstract class, concrete class</p>	<p>Abstract Classes assignment</p> <p>Lesson 2.9 quiz through Edhesive</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Obtain a deeper understanding of complex class hierarchies and inheritance in constructors.</p> <p>Investigate worked examples of AP Computer Science A exam questions on inheritance</p>	<p>Object of class A “is-a” object of class B only if class A inherits from class B</p> <p>Is-A relationships only work in one way</p> <p>When to utilize the Super constructor</p> <p>Vocabulary: Is-A and Has-A relationships, super</p>	<p>Is-A and Has-A Relationships assignment</p> <p>AP practice questions</p> <p>Lesson 2.10 quiz through Edhesive</p>	<p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p> <p>3B-AP-23 Evaluate key qualities of a program through a process such as a code review.</p> <p>A-SSE.3c Choose and produce an equivalent form of an expression to reveal and explain properties of the quantity represented by the expression.</p>
<p>Understand the purpose and behavior of an interface in Java</p> <p>Identify the similarities and differences between interfaces and abstract classes</p>	<p>Techniques for writing an interface and how they differ from an abstract class</p> <p>Interfaces cannot be instantiated</p> <p>Vocabulary: Interface, abstract class, purpose, behavior</p>	<p>Interfaces assignment</p> <p>Lesson 2.11 quiz through Edhesive</p> <p>Assignment 2.4: Fraction Comparable</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p> <p>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).</p> <p>9.3.IT-PRG.6</p>

			Program a computer applications using the appropriate programming language.
<p>Understand why wrapper classes for primitive data types may be useful</p> <p>Investigate code which makes use of the Integer and Double wrapper classes</p>	<p>Proper syntax for adding a wrapper class to a program</p> <p>How to interpret both integer and double class data points</p> <p>Vocabulary: Wrapper class, integer, double, primitive data type</p>	<p>Wrapper Classes assignment</p> <p>Lesson 2.12 coding activities through Edhesive</p> <p>AP Lab 2: Elevens</p> <p>Lesson 2.12 quiz through Edhesive</p> <p>AP Practice</p> <p>Unit 2.2 Exam</p>	<p>3B-AP-14</p> <p>Construct solutions to problems using student-created components, such as procedures, modules and/or objects</p> <p>8.2.12.E.3</p> <p>Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).</p>

Unit 7 Overview
Unit Title: Algorithms - Searching and Sorting
Unit Summary: In this unit, students will review previously learned concepts related to arrays from the first term. After reviewing, students will thoroughly cover the standard algorithms utilized in Javascript programming that will appear on the AP exam. Students will have the opportunity of solving real world problems by performing searching algorithms on both ordered and unordered lists.
Suggested Pacing: 6 lessons
Learning Targets
Unit Essential Questions: <ul style="list-style-type: none"> • What uses can a computer automated algorithm serve? • What are the differences between analyzing data that is sorted in comparison to unsorted?
Unit Enduring Understandings: <ul style="list-style-type: none"> • Sequential and Binary Searches • Selection, Insertion and Merge sort
Evidence of Learning
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets
Summative Assessments: Lesson quizzes 2.13-2.18, Unit Quiz 2.3, Exam 2.3
Alternative Assessments: Assignment 2.5: Game Wheel, Assignment 2.6: Sort Team Directory, Lesson mini coding activities, AP-style multiple choice practice

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLS)
<p>Review the definition of an algorithm and understand the need for algorithms in searching for and sorting data in arrays</p> <p>Analyze the important properties we may wish to measure for algorithms and when these may be more or less important</p>	<p>In order to complete any computational sorting process using an array, an algorithm is required</p> <p>Vocabulary: Algorithm, array, searching, sorting</p>	<p>Algorithms assignment</p> <p>Lesson 2.13 quiz through Edhesive</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p> <p>CRP11 Use technology to enhance productivity.</p>
<p>Review the steps in the Linear Search algorithm</p> <p>Understand the advantages and disadvantages of using the linear search algorithm to find data in an array</p>	<p>How a linear algorithm sorts through data</p> <p>Linear algorithms may be slow for large data sets</p> <p>Vocabulary: Linear algorithm, searching, sorting, data set</p>	<p>Linear Search assignment</p> <p>Lesson 2.14 coding activities through Edhesive</p> <p>Lesson 2.14 quiz through Edhesive</p> <p>Unit 2.3 Quiz</p> <p>Assignment 2.5: Game Wheel</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p> <p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>
<p>Understand how the selection sort algorithm orders an array of data</p> <p>Write code to implement the selection sort algorithm in Java</p> <p>Understand the advantages and disadvantages of using the selection sort algorithm to order the elements of an array</p>	<p>While the selection sort algorithm is the simplest to program, it is also the slowest option</p> <p>Vocabulary: Selection Sort, algorithm, array</p>	<p>Selection Sort assignment</p> <p>Lesson 2.15 coding activities through Edhesive</p> <p>Lesson 2.15 quiz through Edhesive</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p>
<p>Understand how the insertion sort algorithm orders an array of data</p> <p>Understand the advantages and disadvantages of using the insertion sort algorithm to order the elements of an array</p>	<p>The insertion sort algorithm makes comparisons as it searches through data</p> <p>Vocabulary: Insertion Sort, algorithm, array</p>	<p>Insertion Sort assignment</p> <p>AP practice</p> <p>Lesson 2.16 quiz through Edhesive</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p>
<p>Understand how the binary search algorithm finds a value in an ordered array of data</p> <p>Write code to implement the binary search algorithm in Java</p> <p>Understand the advantages and disadvantages of using the binary search algorithm to find data in an array</p>	<p>The binary search algorithm only works for sorted data sets</p> <p>Vocabulary: Binary Search, algorithm, array</p>	<p>Binary Search assignment</p> <p>Lesson 2.17 coding activities through Edhesive</p> <p>Lesson 2.17 quiz through Edhesive</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.</p>
<p>Understand how the merge sort algorithm orders an array of data</p>	<p>While the merge sort is complex to program, it is the fastest method of data sorting</p> <p>Vocabulary:</p>	<p>Merge Sort assignment</p> <p>Lesson 2.18 quiz through Edhesive</p>	<p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p> <p>8.2.12.E.3</p>

Understand the advantages and disadvantages of using the merge sort algorithm to order the elements of an array	Merge sort, algorithm, array	Unit 2.3 Exam Assignment 2.6: Sort Team Directory	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games). 9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.
---	------------------------------	--	---

Unit 8 Overview	
Unit Title: Two-Dimensional Arrays	
Unit Summary: In this unit, students are introduced to the complex coding functions of two-dimensional arrays. As students learn how to program and create 2-D arrays, students will be introduced to new algorithms to utilize while learning how to apply previously learned skills and abilities in working with arrays. Students will have the opportunity to use 2-D arrays to solve real-world problems through the unit assignments and AP Lab activities within this unit.	
Suggested Pacing: 3 lessons	
Learning Targets	
Unit Essential Questions: <ul style="list-style-type: none"> What is the difference in the functionality and structure of a two-dimensional array in comparison to a one-dimensional alternative? What new problems can be solved with a two-dimensional array that were not previously achievable? 	
Unit Enduring Understandings: <ul style="list-style-type: none"> Two-Dimensional Arrays Algorithms on Two-Dimensional Arrays 	
Evidence of Learning	
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets	
Summative Assessments: Lesson quizzes 2.19-2.21, Unit Quiz 2.4, Exam 2.4	
Alternative Assessments: Assignment 2.7: Battleship, Lesson mini coding activities, AP-style multiple choice practice, AP Lab	

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLS)
<p>Understand how the two-dimensional array structure in Java may be represented by a table with rows and columns</p> <p>Learn how to declare and initialize and access data from a two-dimensional array in Java</p>	<p>How 2-D arrays can process more data and different forms of data in comparison to standard arrays</p> <p>Vocabulary: Two-Dimensional Arrays, table of data, array of arrays</p>	<p>2-D Arrays assignment</p> <p>Lesson 2.19 quiz through Edhesive</p> <p>Unit 2.4 Quiz</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>A-CED.2 Create equations in two or more variables to represent relationships between quantities; graph equations on coordinate axes with labels and scales.</p> <p>CRP11 Use technology to enhance productivity.</p>
<p>Understand how common algorithms for 2-D arrays are implemented</p> <p>Write a method which implements an algorithm for processing data in a 2-D array</p>	<p>What real-world data calculations can be completed using 2-D arrays</p> <p>Vocabulary: Two-Dimensional Arrays, method, algorithm</p>	<p>2-D Array Algorithms assignment</p> <p>Lesson 2.20 coding activities through Edhesive</p> <p>Lesson 2.20 quiz through Edhesive</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p> <p>8.2.2.E.4 Debug an algorithm (i.e., correct an error).</p>
<p>Practice answering an AP Computer Science A free response question involving writing an entire class</p> <p>Analyze a worked solution for this question and understand how a solution would be marked</p>	<p>How AP free response questions are scored</p> <p>Techniques for declaring classes and methods on a free response format</p> <p>Vocabulary: Free response, class, method, constructor, parameter</p>	<p>Tracing Code assignment</p> <p>Lesson 2.21 coding activities through Edhesive</p> <p>Lesson 2.21 quiz through Edhesive</p> <p>AP Lab 3: Picture Lab</p> <p>Unit 2.4 Exam</p> <p>Assignment 2.7: Battleship</p>	<p>3B-AP-23 Evaluate key qualities of a program through a process such as a code review.</p> <p>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).</p> <p>9.3.IT-PRG.6 Program a computer applications using the appropriate programming language.</p>

Unit 9 Overview	
Unit Title: AP Exam Review	
Unit Summary: In this unit, students will review all previously learned content covered in this course in order to prepare them for the AP Computer Science A exam. Students will be given a diagnostic exam and practice AP problems to prepare them to take the AP Computer Science exam. A series of review videos are provided covering the major topics on the AP Exam. The goal of the unit is to allow students to synthesize the material covered throughout the year and review any areas that could use strengthening.	
Suggested Pacing: 8 lessons	
Learning Targets	
Unit Essential Questions: <ul style="list-style-type: none"> • What are the core elements of the Javascript computer programming language? • What are the skills and techniques required to write code by hand? • What connections and patterns can be made between previously learned skills and content in writing code? • How can code be used to solve everyday problems? 	
Unit Enduring Understandings: <ul style="list-style-type: none"> • Code is repetitive, patterns exist between all forms of syntax • The ability to write code by hand without a computer is one that strengthens computational abilities • Computer programming is intertwined in nearly every product, industry, and progression in our modern lives 	
Evidence of Learning	
Formative Assessments: A variety of formative assessments will be used throughout the lesson, such as mini code challenges, group discussions, exit tickets	
Summative Assessments: Diagnostic Exam, review quizzes	
Alternative Assessments: AP-style multiple choice practice	

Objectives (Students will be able to...)	Key Concepts (Students will know...)	Suggested Assessments	Standards (NJSLS)
<p>Use previously learned skills and abilities in writing Java to answer questions for solving real-world problems</p> <p>Identify their strengths and weaknesses with reading and writing the Java coding language</p>	<p>The structure of the AP CS A Exam</p> <p>Vocabulary: Programming Fundamentals, Data Structures, Logic, Algorithms, Object Oriented Programming, Recursion, Software Engineering</p>	<p>AP Diagnostics Exam</p> <p>AP practice questions</p>	<p>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).</p> <p>CRP2 Apply appropriate academic and technical skills.</p>
<p>Identify fundamental syntax and programming functions needed to create a Java application</p> <p>Create a program using Javascript coding skills</p>	<p>The fundamental concepts of the Javascript programming language</p> <p>Vocabulary: Data types, numeric casts, modular division, math functions, while loops, for loop, parameters primitive vs class, overloaded methods, arrayList, for-each loop, interfaces</p>	<p>Programming Fundamentals review quiz</p> <p>AP practice questions</p>	<p>3B-AP-12 Compare and contrast fundamental data structures and their uses.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Review the proper syntax and coding methods for previously learned skills and abilities in working with Data Structures in Java</p> <p>Create a program using Javascript coding skills</p>	<p>The methods and techniques needed to program Data Structures in Javascript programming applications</p> <p>Vocabulary: Tracing code, string functions, one-dimensional arrays, two-dimensional arrays, arrayList, arrays of strings</p>	<p>Data Structures review quiz</p> <p>AP practice questions</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>3B-AP-21 Develop and use a series of test cases to verify that a program performs according to its design specifications.</p>
<p>Review the proper syntax and coding methods for previously learned skills and abilities in working with Logic functions and decision making in Java</p> <p>Create a program using Javascript coding skills</p>	<p>The methods and techniques needed to program Logic and autonomous decision making in Javascript programming applications</p> <p>Vocabulary: Boolean, operators, truth tables, DeMorgan's law, math functions, Ifs, else</p>	<p>Logic review quiz</p> <p>AP practice questions</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>
<p>Review the proper syntax and coding methods for previously learned skills and abilities in working with the various forms of Algorithms in Java</p> <p>Create a program using Javascript coding skills</p>	<p>The methods and techniques needed to program the various forms of Algorithms in Javascript programming applications</p> <p>Vocabulary: DeMorgan's Law, tracing code, algorithms, parameters, primitive vs class, return methods, constructors</p>	<p>Algorithms review quiz</p> <p>AP practice questions</p>	<p>3B-AP-10 Use and adapt classic algorithms to solve computational problems.</p> <p>3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.</p>
<p>Identify the core fundamental principles with object oriented programming systems</p> <p>Review the proper syntax and coding methods for previously learned skills and abilities in working with the object-oriented structure of Java</p>	<p>Understand the fundamental principles with object-oriented programming structures</p> <p>Vocabulary: Inheritance, Overriding, abstract, Is-A, Has-A, interfaces</p>	<p>Object-Oriented programming review quiz</p> <p>AP practice questions</p>	<p>3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p> <p>3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.</p>

<p>Review the proper syntax and coding methods for previously learned skills and abilities in working with the fundamentals of recursion techniques in Java</p> <p>Create a program using Javascript coding skills</p>	<p>The methods and techniques needed to program the fundamentals of recursion techniques in Javascript programming applications</p> <p>Vocabulary: Modular Division, Recursion, Recursive functions, Returns</p>	<p>Recursion review quiz</p> <p>AP practice questions</p>	<p>3B-AP-21</p> <p>Develop and use a series of test cases to verify that a program performs according to its design specifications.</p>
<p>Recognize code that correctly follows engineering principles</p> <p>Create a program using Javascript coding skills</p>	<p>The methods and techniques needed to program using engineering principles in Javascript programming applications</p> <p>Vocabulary: Inheritance, abstract classes, Is-A, Has-A, interfaces</p>	<p>Software Engineering review quiz</p> <p>AP practice questions</p>	<p>3B-AP-14</p> <p>Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</p>