ROBBINSVILLE PUBLIC SCHOOLS

OFFICE OF CURRICULUM AND INSTRUCTION

DEPARTMENT
Technology

COURSE TITLE
AP Computer Science A

Board of Education

Curriculum Writing Committee

Radhika Vaidyanathan

Supervisors

Tiffany Brennan

BOARD OF EDUCATION INITIAL ADOPTION DATE:

Course Philosophy

AP Computer Science A program is for students, after taking Introduction to Computer Science courses, find their passion in Computer Science. This college level program course challenges students. Students enter AP Computer Science A with varying backgrounds. In order to meet the diverse needs and appeal to the interests of my students, I try whenever possible to teach concepts inside of contexts. This keeps students motivated and allows me to spiral my instruction. Working with contexts allows me the chance to arrange topics to maximize student excitement at important times, such as the beginning and end of first semester, and just before students choose their courses for the following year.

Course Description

Course Prerequisite: Computer Science II *Full Year (5.0 credits)* *Grade: 10-12*

AP Computer Science A introduces students to computer science through programming. Fundamental topics in this course include the design of solutions to problems, the use of data structures to organize large sets of data, the development and implementation of algorithms to process data and discover new information, the analysis of potential solutions, and the ethical and social implications of computing systems. The course emphasizes object-oriented programming and design using the Java programming language.

Core and Supplemental Instructional Materials

Core Materials	Supplemental Materials
● Java Methods: OOP and Data Structures (Third Edition) by Maria Litvin / Gary Litvin.	● Progress checks, daily videos, and practice tests from AP Classroom ● RuneStone Academy

	● CodeHS.com

Social Emotional Learning Connections

Below are the five core SEL Competencies as outlined by CASEL, and examples of how each may be addressed within this curriculum

Self-awareness: The ability to accurately recognize one’s emotions and thoughts and their influence on behavior. This includes accurately assessing one’s strengths and limitations and possessing a well-grounded sense of confidence and optimism.

Example 1: Students explore how awareness of one’s strengths and limitations prepare them to take an AP level course.
Example 2: Computer science can help students as they develop their self-awareness. Coding includes lots of trial and error and debugging, and we can encourage our students to develop their self-awareness through reinforcing a growth mindset, providing ample opportunities for reflection, identifying emotions and thoughts about successes and challenges with coding, and tying students' interests and identities into their coding projects.

Self-management: The ability to regulate one’s emotions, thoughts, and behaviors effectively in different situations. This includes managing stress, controlling impulses, motivating oneself, and setting and working toward achieving personal and academic goals.

Example 1: Self-management means that kids use skills and mindsets to create and persevere toward their goals
Example 2: Students practice self-management by taking initiative, planning, and organizing the steps needed to solve the given problem.

Social awareness: The ability to take the perspective of and empathize with others from diverse backgrounds and cultures, to understand social and ethical norms for behavior, and to recognize family, school, and community resources and supports.

Example 1: Students who are programming computers for others, need to understand the different background and culture where the software is being used for.
Example 2: Students develop and plan a program, brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users, and evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

Relationship skills: The ability to establish and maintain healthy and rewarding relationships with diverse individuals and groups. This includes communicating clearly, listening actively, cooperating, resisting inappropriate social pressure, negotiating conflict constructively, and seeking and offering help when needed.

Example 1: Students also collaborate with other students on pair programming, small group projects, and peer feedback.
Example 2: Students practice and develop teamwork, collaborative problem-solving, and constructive ways to resolve conflicts and differences of opinion or approach.

Responsible decision-making: The ability to make constructive and respectful choices about personal behavior and social interactions based on consideration of ethical standards, safety concerns, social norms, the realistic evaluation of consequences of various actions, and the well-being of self and others.

Example 1: Responsible decision making is a key part of Computer Science programming because of the widerimplications of AI and machine learning.
Example 2: Students develop decision making skills through collecting and analyze data, along with thinking through positive and negative implications of technologies.

Integration of 21st Century Themes and Skills

NJSLS-CLKS 9.4: Life Literacies and Key Skills	
Creativity and Innovation	*See specific standards and their connections/examples for this disciplinary concept listed within each individual unit* Can be found in unit: 5 and 7
Critical Thinking and Problem Solving	*See specific standards and their connections/examples for this disciplinary concept listed within each individual unit* Can be found in unit: 1, 2, 3
Digital Citizenship	*See specific standards and their connections/examples for this disciplinary concept listed within each individual unit* Can be found in unit: 5 and 7
Global and Cultural Awareness	*See specific standards and their connections/examples for this disciplinary concept listed within each individual unit* Can be found in unit: 5 and 7
Information and Media Literacy	*See specific standards and their connections/examples for this disciplinary concept listed within each individual unit* Can be found in unit: 6,7,8
Technology Literacy	*See specific standards and their connections/examples for this disciplinary concept listed within each individual unit* Can be found in unit: All Units

Robbinsville Ready 21st Century Skill Integration

The following skills will be embedded throughout the curriculum and instruction of this course.

Collaborative Team Member: Robbinsville students will learn more by working together than in isolation. As educational theorist Lev Vygotsky advocated, learning is a social process. Many workplaces today encourage employees to work in teams to solicit diverse perspectives, brainstorm new ideas and/or products, and solve problems. Further, collaboration fosters interpersonal relationships, self-management skills, cooperation, and a sense of collective responsibility. Collaborative team members are able to work with diverse groups of people who hold a variety of perspectives.

Effective Communicator: Robbinsville students must be able to clearly articulate their ideas orally, in writing, and across various media in order to successfully connect to the world around them. As the world becomes increasingly globalized, communication is more than just sharing one's ideas. Effective communicators are able to communicate their convictions, actively listen and analyze others' work to identify perspective and/or potential bias.

Emotionally Intelligent Learner: Robbinsville students who are emotionally intelligent learn to be empathetic, demonstrate integrity and ethical behavior, are kind, are self-aware, willing to change, and practice self-care. They are better able to cope with the demands of the 21st century digital society and workplace because they are reliable, responsible, form stable and healthy relationships, and seek to grow personally and professionally. Emotionally intelligent people are able to manage their emotions, work effectively on teams and are leaders who can grow and help to develop others.

Informed and Involved Citizen: Robbinsville students need to be digital citizens who are civically and globally aware. The concept of what it means to be "literate" has evolved along with 21st century technological and cultural shifts. Our progressive vision of literacy entails having our students explore real world problems in the classroom. Informed and involved citizens are able to safely and accurately communicate with people all around the world and are financially, environmentally and informationally literate.

Innovative Thinker: Robbinsville students must encompass innovative thinking skills in order to be successful lifelong learners in the 21st century world. As stated by Karl Fisch and Scott McLeod in the short film Shift Happens, "We are currently preparing students for jobs that don't yet exist . . . using technologies that haven't been invented . . . in order to solve problems we don't even know are problems yet." Innovative thinkers are able to think analytically, solve problems critically, creatively engage in curiosity and tinkering, and demonstrate originality.

Resilient and Self-Directed Learner: Robbinsville students need to take risks and ultimately make independent and informed decisions in an ever-changing world. Author of Life, the Truth, and Being Free, Steve Maraboli stated, “Life doesn’t get easier or more forgiving, we get stronger and more resilient.” Self-directed scholars of the 21st century are able to set goals, initiate resolutions by seeking creative approaches, and adjust their thinking in light of difficult situations. Resilient students are able to take risks without fear of failure and overcome setbacks by utilizing experiences to confront new challenges. Resilient and self directed scholars will consistently embrace opportunities to initiate solutions and overcome obstacles.

Career Awareness and Planning Standards 9.2	
9.2.12.C.2: Modify Personalized Student Learning Plans to support declared career goals.	As students learn programming languages, they are exploring an opportunity in the field of Computer Science. They modify their learning plan to support their career in Computer Science.
9.2.12.C.7: Examine the professional, legal, and ethical responsibilities for both employers and employees in the global workplace.	Students learn about ethical responsibilities with user's data when learning Unit 6,7 and 8.
9.2.12.C.4: Analyze how economic conditions and societal changes influence employment trends and future education.	Students learn Computer Science as it is an integral part of the future. Data security is a major concern and students learn in this course how to take that into consideration when dealing with user's data.

Robbinsville Public Schools
Scope, Sequence, Pacing and Assessment

AP Computer Science A

Unit Title	Unit Understandings and Goals	Recommended Duration/ Pacing	Assessments
Unit 1: Primitive Types	- Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence. - To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values. - The way variables and operators are sequenced and combined in an expression determines the computed result.	4 - 6 Class periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 2: Using Objects	- Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence. - To find specific solutions to generalizable problems, programmers include variables	9 - 11 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing

	in their code so that the same algorithm runs using different input values.		· End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 3: Boolean Expressions and if Statements	- The way variables and operators are sequenced and combined in an expression determines the computed result. - Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	8 - 10 Class periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 4: Iteration	- Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	12 - 14 Class periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects

			· Students' presentations
Unit 5: Writing Classes	- Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept. - When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses. - To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values. - While programs are typically designed to achieve a specific purpose, they may have unintended consequences.	10 - 12 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 6: Array	- To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value. - Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	6 – 8 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 7: ArrayList	- To manage large amounts of data or complex relationships in data, programmers write code that groups the	8 - 10 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions

	data together into a single data structure without creating individual variables for each value. - Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values. - While programs are typically designed to achieve a specific purpose, they may have unintended consequences.		· Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 8: 2D Array	- To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value. - To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.	8 - 10 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 9: Inheritance	- When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.	13 – 15 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz

			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations
Unit 10: Recursion	- Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	3 – 5 Class Periods	Formative · Do-Now / Exit Tickets · Poll Questions · Labs for each unit · Classwork / Homework submissions
			Summative · Teacher feedback · Peer editing · End of Unit Test/Quiz
			Common Benchmark Assessments (mid/end of course) · Mid-Term Exam and End of Course Project
			Alternative Assessments (projects, etc when appropriate) · Group Projects · Students' presentations

Unit # 1: Primitive Types

Enduring Understandings:

- Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.
- To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.
- The way variables and operators are sequenced and combined in an expression determines the computed result.

Essential Questions:

- How can we use programs to solve problems?
- In what ways are numbers used in the programs and apps you use most often?
- How are mathematical concepts being used in the programs and apps that you use most often?

Interdisciplinary Connections

A-SSE: Seeing Structure in Expressions: Students use Java programming language to solve arithmetic expressions.
A -APR: Arithmetic with Polynomials and Rational Expressions: Students use programming languages to solve polynomial identities.

Career/Real World Connections

Example: Graphing Calculator: Students will develop a graphing calculator that uses different expressions for calculations.

Guiding / Topical Questions with Specific Standards		**Content, Themes, Concepts, and Skills**	**Teaching Strategies**	**Instructional Resources and Materials**	**Assessment Strategies**
8.1.8 .AP.2	**Why Programming? Why Java?**	Call System class methods to generate output to the console. Create string literals.	Do Now Teacher instruction (live coding) on System.out.print and println statements Error analysis Practice problems from the Java Methods book.	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Peer Feedback Project Rubric Lab work
8.1.8 .AP.2	**Variables and Data Types**	Identify the most appropriate data type category for a particular specification.	Do Now		

		Declare variables of the correct types to represent primitive data.	Group Activity on different types of data types Guided practice on using variables and data types Independent practice (Exercise questions) MCQ practice in RuneStone Academy and AP Classroom.		
8.1.8 .AP.2	**Expressions and Assignment Statements**	Evaluate arithmetic expressions in a program code.	Do Now: Debugging Teacher instruction (live coding) on using arithmetic operators in Java. Activating prior knowledge by having students apply what they know from Math to solve an expression and test in a program. MCQ practice questions from AP Classroom		
8.1.8 .AP.2	**Compound Assignment Operators**	Evaluate what is stored in a variable as a result of an expression with an assignment statement.	Do Now: Shortcut to using arithmetic operators **Sharing and responding:** Students in a group of two work on 2 different sets of statements. One set contains compound assignment and other set the same arithmetic expressions without using compound assignment. Students take turns in figuring out what each compound assignment does. Practice questions from Java methods book Practice lessons from RuneStone Academy		

8.1.8.AP.2	**Casting and Ranges of Variables**	Evaluate arithmetic expressions that use casting.	Do Now: Statements given to students that do not use casting Live Coding with teacher instruction on what casting is and how to use it in programs so that you avoid logical errors in the program. **Predict and compare:** Students are given a different set of code segments. They are asked to predict the output for the given segments. Compare the results with the seat partner. Unit 1 practice MCQ and FRQ from AP Classroom Unit 1 practice from RuneStone Academy.		

Unit # 2: Using Objects

Enduring Understandings:	Essential Questions:
● Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence. ● To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.	● How can we simulate election results using existing program code? ● How are appropriate variables chosen to represent a remote control? ● How do the games we play simulate randomness?

Interdisciplinary Connections

F-BF: Build a function that models a relationship between two quantities
Students use the same concept of Functions in Math to write methods in Java that take values in and give out the result at the end.

F-BF: Build new functions from existing functions
Students will also learn how to use the existing methods to write new methods that use them. This avoids ambiguity.

Career/Real World Connections

Example: Library System: Students will create a library system that has various attributes and behaviors.
Student Records: Students will discuss what attributes and behaviors that high school students' records have and build a system where they can store different students' records.

Guiding / Topical Questions with Specific Standards		Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies
8.1.8.A P.4 8.1.8.A P.5	**Objects: Instances of Classes**	Explain the relationship between a class and an object.	Do Now: What is a class in Java? **Using manipulatives:** Students are introduced to objects in Java using hands-on activities. Play-Doh: Cookie cutter in play-doh is used to explain class and when students use the cutter	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource	Teacher Feedback from classwork Peer Feedback Project Rubric

			to cut shapes from the play-doh are treated as objects. Practice questions in creating test classes for the given class from Java Methods book	AP Classroom from CollegeBoard Website	Lab work
8.1.8.AP.4 8.1.8.AP.5	**Creating and Storing Objects (Instantiation)**	Identify, using its signature, the correct constructor being called. For creating objects: a. Create objects by calling constructors without parameters. b. Create objects by calling constructors with parameters. Define variables of the correct types to represent reference data.	Do Now: What are objects? **Marking the text:** Students are given instances of objects in a test class along with class. Students will be working in pairs in identifying keywords used in creating the instances. Live coding with teacher instruction on how to initialize variables when creating objects for the given class. MCQ practice questions from AP Classroom Practice problems from Java Methods book		
8.1.8.AP.4 8.1.8.AP.5	**Calling a Void Method**	Call non-static void methods without parameters.	Do Now: What are methods? How similar are they with functions? Live coding with teacher instruction on how to write methods inside a class. Students discuss the options that they must use to call the method from the test class. FRQ practice questions from AP Classroom Practice lesson from RuneStone Academy		

8.1.8.AP.4 8.1.8.AP.5	**Calling a Void Method with Parameters**	Call non-static void methods with parameters.	Do Now: What are methods? How similar are they with functions? Live coding with teacher instruction on how to write methods inside a class. Students discuss the options that they must use to call the method from the test class. FRQ practice questions from AP Classroom Practice lesson from RuneStone Academy		
8.1.8.AP.4 8.1.8.AP.5	**Calling a Non-void Method**	Call non-static non-void methods with or without parameters.	Do Now: What are methods? How similar are they with functions? Live coding with teacher instruction on how to write methods inside a class. Students discuss the options that they must use to call the method from the test class. FRQ practice questions from AP Classroom Practice lesson from RuneStone Academy		
8.1.8.AP.4 8.1.8.AP.5	**String Objects: Concatenation, Literals, and More**	For String class: a. Create String objects. b. Call String methods.	Do Now: How is String different from primitive data types? Live coding and class discussion on how to create String variables. MCQ practice on how the String values and variables are used in Java programs. Practice questions from Java Methods book Lab work: Teen Talk A statement from the user is taken and added “like” between each word.		

8.1.8.A P.4 8.1.8.A P.5	**String Methods**	For String class: a. Create String objects. b. Call String methods.	Do Now: Watch edpuzzle video on String Methods Small Group work: Students take turns in trying out different String methods from the Java Methods books that explains the behavior of each method. FRQ practice questions from AP Classroom Labwork: Cooney Likes Students understand the game as we play in class. After they have figured out the game, they complete the given program with the missing code.		
8.1.8.A P.4 8.1.8.A P.5	**Wrapper Classes: Integer and Double**	For wrapper classes: a. Create Integer objects. b. Call Integer methods. c. Create Double objects. d. Call Double methods.	Do Now: What is the difference between int, Integer, double and Double? Live coding and teacher instruction on how to use wrapper classes Integer and Double. MCQ practice questions from AP Classroom Practice lesson from RuneStone Academy.		
8.1.8.A P.4 8.1.8.A P.5	**Using the Math Class**	Call static methods from Math Class. Evaluate expressions that use the Math class methods.	Do Now: How to find the exponent in Java? Students are introduced to Math Methods that are available in Java. Live coding with teacher instruction on what different Math methods are available. Think-Pair-Share: Students are given different sets of questions that use		

			different Math methods in solving a problem. Students trace the code in finding out the answer to each question and discuss them in pairs. RuneStone Academy practice questions from Unit 2 Unit 2 Progress checks (MCQ and FRQ) from AP Classroom		

Unit # 3: Boolean Expressions and if Statements

Enduring Understandings:	Essential Questions:
● The way variables and operators are sequenced and combined in an expression determines the computed result. ● Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values. ●	● How can you use different conditional statements to write a pick-your-own-path interactive story? ● Why is selection a necessary part of programming languages?

Interdisciplinary Connections

S-CP: Conditional Probability and the Rules of Probability
Students will use "and", "or", "not" in many circumstances of conditional statements to write a solution to the given question.

S-MD: Using Probability to Make Decisions
Students will use random number generation to play games and also conditional statements to make decisions according to the user given input.

Career/Real World Connections

Example: Take 123 Game: Students will build a game using conditional statements that needs to check for the number of stones left after each turn.
Income Tax calculator: Based on the user income and deductions, the system will be built to calculate the tax the customer owes to the IRS.

Guiding / Topical Questions with Specific Standards		Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies
8.1.8.AP.3	**Boolean Expressions**	Evaluate Boolean expressions that use relational operators in program code.	Do Now: Decisions that you make everyday Group Discussion: Students discuss how decisions affect the program execution.	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource	Teacher Feedback from classwork Peer Feedback Project Rubric Lab Work

			Live coding with teacher instruction on Boolean expressions in Java. Practice questions from the Java Methods book.	AP Classroom from CollegeBoard Website	
8.1.8.AP.3	**if Statements and Control Flow**	Represent branching logical processes by using conditional statements.	Pair Programming: Students take turns in analyzing the conditional statements given in the code segments. Live coding instruction on learning conditional statements in Java. Flowchart: Students use online diagram tools or paper to draw flowchart for the given code segment. Practice problems from lesson 3.2 RuneStone Academy Practice problems from the Java Methods book.		
8.1.8.AP.3	**if-else Statements**	Represent branching logical processes by using single conditional statements.	Do Now: Students discuss what happens when the condition is not satisfied in the given code segment. **Code tracing:** Students are provided with several code segments that contain conditional statements. They will trace the given sample inputs and predict the output for each. Practice questions (MCQ) from AP Classroom Practice problems from the Java Methods book.		
8.1.8.AP.3	**else if Statements**	Represent branching logical processes by using cascaded conditional statements.	Do Now: Students discuss what happens when the condition is not satisfied in the given code segment. **Code tracing:** Students are provided with several code segments that contain conditional statements. They		

			will trace the given sample inputs and predict the output for each. Practice questions (MCQ) from AP Classroom Practice problems from the Java Methods book.		
8.1.8.AP.3	**Compound Boolean Expressions**	Represent branching logical processes by using nested conditional statements. Evaluate compound Boolean expressions in program code.	Do Now: T chart on "And", "Or" and "Not" **Diagramming:** Students create truth tables by listing all possible true and false combinations and corresponding Boolean values for the given compound Boolean expressions. Live Coding with teacher instruction on how to combine Boolean expressions in Java. Short Circuit evaluation: Students are given code segments that may produce errors. Students discuss how to fix those errors. Practice FRQ SpinnerGame from AP Classroom Practice problems using Java Methods book.		
8.1.8.AP.3	**Equivalent Boolean Expressions**	Compare and contrast equivalent Boolean expressions.	Student response system: Students are provided with a code segment that utilizes conditional statements and a compound Boolean expression, and they are asked to choose an equivalent code segment that uses a nested conditional statement (and vice versa). They report their responses using a student response system. Take123 Game: Students are introduced to the game and given		

			tokens. As a group, students play the game with each other to understand the strategy behind each play. After understanding the strategy, they work on completing the program that plays the game between the user and the computer.		
8.1.8.AP.3	**Comparing Objects**	Compare object references using Boolean expressions in program code.	Do Now: Comparing String and other objects. **Predict and compare:** Live Coding with teacher instruction on how to compare String and other objects. Students participate in predicting the output of several different code segments. After predicting, they write programs and get hands-on experience using the .equals() method for Strings and objects. RuneStone Academy practice questions from Unit 3 Unit 3 Progress checks (MCQ and FRQ) from AP Classroom		

Unit # 4: Iteration

Enduring Understandings:	**Essential Questions:**
● Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	● How does iteration improve programs and reduce the amount of program code necessary to complete a task? ● What situations would warrant the use of one type of loop over another?

Interdisciplinary Connections

A-SSE: Write expressions in equivalent forms to solve problems
Students use iteration in Java to solve an arithmetic and geometric series.

A -APR: Use polynomial identities to solve problems
Students use binomial theorem and Pascal triangle series from Math to implement them using Java.

Career/Real World Connections

Example: Coke has experimented with different formulations of its popular Coca Cola product, including the failed New Coke iteration. The aim is to achieve the best tasting product that customers like.
A cook may experiment with a recipe, tweaking the ingredients or changing different steps of the process slightly until the food tastes as good as it possibly can.

Guiding / Topical Questions with Specific Standards		**Content, Themes, Concepts, and Skills**	**Teaching Strategies**	**Instructional Resources and Materials**	**Assessment Strategies**
8.1.8.AP.3:	**while Loops**	Represent iterative processes using a while loop. For algorithms in the context of a particular specification that does not require the use of traversals: 1. Identify standard algorithms. 2. Modify standard algorithms.	Do Now: Limitations of conditional statements in some cases. Live coding with teacher instructions on user input validation using a while loop.	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource	Teacher Feedback from classwork Peer Feedback Project Rubric

		3. Develop an algorithm.	As a group, students are given different examples of code	AP Classroom from CollegeBoard Website	Lab work
8.1.8.AP.3:	**for Loops**	Represent iterative processes using a for loop.	Do Now: T-Chart for difference between while and for loop. Live coding with teacher instruction on how to use for loops in Java. Student discussion on differences between while and for loop. **Jigsaw:** As a whole class, students are given a code segment with iteration and its output. Then, students are divided into groups and given a slightly modified code segment. As a group they work together to determine how the result changes based on the modified segment. They share their conclusion on what their output is for the modified code segment. Practice MCQ questions from AP classroom Practice problems from the Java Methods book.		
8.1.8.AP.3: 8.1.12.AP.5	**Developing Algorithms Using Strings**	For algorithms in the context of a particular specification that involves String objects: a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Do Now: FRQ practice from AP classroom **Note-taking:** Students are provided with a method that, when given an integer, returns the month name from a String that includes all the month names in order, each separated by a space. They are going to annotate what each statement does in the method. Then, use their annotated method as a guide to write a similar method that, given a student number as input, returns the name of a student from a String containing the first name of all		

			students in the class, each separated by a space. Practice questions from the Java Methods book.		
8.1.8.AP.3:	**Nested Iteration**	Represent nested iterative processes.	Do Now: Correct the indentation given in the code segment. Live coding and teacher instruction on introducing nested loops. Students are given practice questions (MCQ) on a nested loop and have them trace the loop to find the answer to the given questions. Lab work: Perfect Numbers Students work individually to complete the given code segment that prints the first five perfect numbers starting from 0.		
	Informal Code Analysis	Compute statement execution counts and informal run-time comparison of iterative statements.	Do Now: Practice MCQ questions from AP Classroom **Simplify the problem:** Students are given several code segments that contain iteration. Students practice tracing through each question and finding out how many times the statements are executed inside the loop and what is the outcome of the given problem. RuneStone Academy practice questions from Unit 4 Unit 4 Progress checks (MCQ and FRQ) from AP Classroom		

Unit # 5: Writing Classes

Enduring Understandings:	**Essential Questions:**
● Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept. ● When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses. ● To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values. ● While programs are typically designed to achieve a specific purpose, they may have unintended consequences.	● How can using a model of traffic patterns improve travel time? ● How can programs be written to protect your bank account balance from being inadvertently changed? ● What responsibility do programmers have for the consequences of programs they create, whether intentional or not?

Interdisciplinary Connections

6.SP: Develop understanding of statistical variability.
Students use data collection in creating a class with certain attributes and behaviors

G-GPE: Expressing Geometric Properties with Equations
Students use different geometric properties in translating it in a Java class.

Career/Real World Connections

Example: Consider a car as a class that has characteristics like steering wheels, seats, brakes, etc. And its behavior is mobility. But we can say Honda City having a registration number 4654 is an ‘object’ that belongs to the class ‘car’.

The High School Student system is a class that can have many attributes and behaviors that correspond to a student in high school.

Guiding / Topical Questions with Specific Standards	Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies

8.1.12. AP.5 8.1.12. AP.6	**Anatomy of a Class**	Designate access and visibility constraints to classes, data, constructors, and methods. Designate private visibility of instance variables to encapsulate the attributes of an object.	Do Now: Students' ideas of what a class is. **Kinesthetic learning:** In a small group, students play a board game. As they play for 10 minutes, they write down all the attributes(nouns) and behaviors(actions) in the game. With this data that they have created, they are introduced, in a live coding lesson, to what a class is in Java and how attributes and behaviors are a part of a class. Practice questions from Java Methods book and RuneStone Academy 5.1	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Peer Feedback Project Rubric Lab Work
8.1.12. AP.5 8.1.12. AP.6	**Constructors**	Define instance variables for the attributes to be initialized through the constructors of a class.	Do Now: What are the different things in a class that you notice? Students are introduced to different sections of a class. Students as a group identify different sections in each given class. Practice questions for writing constructors for the concept discussed from last class. Practice: Write program code to define a new type by creating a class.		
8.1.12. AP.5 8.1.12. AP.6	**Documentation with Comments**	Describe the functionality and use of program code through comments.	Do Now: What are comments in a Java program? Live coding with teacher instruction on introducing the comments for the section of the class. **Marking the text:** Students are given specifications and they highlight what are preconditions for each method		

			given in the class. They also include information about parameters and return value of each method. Practice questions from RuneStone Academy.		
8.1.12. AP.5 8.1.12. AP.6	**Accessor Methods**	Define behaviors of an object through non-void methods without parameters written in a class.	Do Now: Review methods in java program. (5.4 to 5.7) **Create a plan**: Students are asked to outline using pseudocode with paper and pencil. Go through it step-by-step with sample input to ensure that the process is correct and to determine if any additional information is needed before starting to write the program on the computer. Introduce students to different security measures to be followed in writing a class. Practice questions from AP Classroom.		
8.1.12. AP.5 8.1.12. AP.6	**Mutator Methods**	Define behaviors of an object through void methods with or without parameters written in a class.	Do Now: Review methods in java program. (5.4 to 5.7) **Create a plan**: Students are asked to outline using pseudocode with paper and pencil. Go through it step-by-step with sample input to ensure that the process is correct and to determine if any additional information is needed before starting to write the program on the computer. Introduce students to different security measures to be followed in writing a class.		

			Practice questions from AP Classroom.		
8.1.12. AP.5 8.1.12. AP.6	**Writing Methods**	Define behaviors of an object through non-void methods with parameters written in a class.	Do Now: Review methods in java program. (5.4 to 5.7) **Create a plan**: Students are asked to outline using pseudocode with paper and pencil. Go through it step-by-step with sample input to ensure that the process is correct and to determine if any additional information is needed before starting to write the program on the computer. Introduce students to different security measures to be followed in writing a class. Practice questions from AP Classroom.		
8.1.12. AP.5 8.1.12. AP.6	**Static Variables and Methods**	Define behaviors of a class through static methods. Define the static variables that belong to the class.	Do Now: T-Chart between static and non-static variables. **Paraphrase:** Students are given several examples of static and non-static variables and methods. As a group, they are going to categorize them in the given chart. After categorizing them, students are going to write a program using static and non-static variables and methods. Practice questions from Java Methods book Practice problems from RuneStone Academy.		

8.1.12. AP.5 8.1.12. AP.6	**Scope and Access**	Explain where variables can be used in the program code.	Do Now: Find errors in the given classes Live Coding with teacher instruction on how to access variables and methods from outside the class. Practice questions from Java Methods Book.		
8.1.12. AP.5 8.1.12. AP.6	**this Keyword**	Evaluate object reference expressions that use the keyword this.	Do Now: Same variable names for parameter and instance variable. Discussion: Students discuss as a group how to solve problems when using the same variable names for parameters and instance variables. Live coding and teacher instruction on how to use the keyword "this" in a single class. Lab work: Soccer Tournament Students as a group, categorize each variable and method needed for the given challenge to create Soccer Teams and have a tournament between them. After categorizing the variables and methods, they work on writing programs and completing the given task. RuneStone Academy practice questions from Unit 5 Unit 5 Progress checks (MCQ and FRQ) from AP Classroom		
8.2.12. EC.3	**Ethical and Social Implications of Computing Systems**	Explain the ethical and social implications of computing systems.	Do Now: Share the risk of sharing personal details online. **Discussion group:** Students research risks to privacy from collecting and		

			storing personal data on computer systems. Presentation: Students work on creating a presentation on what is privacy and how to protect your personal information online. Play Google Developed game of InterLand.		

Unit # 6: Array

Enduring Understandings:	**Essential Questions:**
● To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value. ● Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	● How can programs leverage volcano data to make predictions about the date of the next eruption? ● How can knowing standard algorithms be useful when solving new problems?

Interdisciplinary Connections

S-ID: Summarize, represent, and interpret data on a single count or measurement variable
Summarize, represent, and interpret data on two categorical and quantitative variables

Example: Students analyze and interpret data in an array or arraylist. Collection of temperatures ina given month or collections of grades of a student in a high school are some examples of data.

S-IC: Understand and evaluate random processes underlying statistical experiments
Make inferences and justify conclusions from sample surveys, experiments, and observational studies

Example: Students use the data to analyze different sorting and searching algorithms and their efficiency.

Career/Real World Connections

Example: List of transactions for credit cards. Studentswill store the records of transactions in an array.
List of students from a high school.

Guiding / Topical Questions with Specific Standards	Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies

8.1.12. AP.2	**Array Creation and Access**	Represent collections of related primitive or object reference data using one- dimensional (1D) array objects.	Do Now: Con of using single variable Introduction to Array PearDeck Activity with interactive slides Live Coding with teacher instruction on introduction to array. **Diagramming** : Students are provided with several prompts to create and access elements. Students draw a memory diagram that shows references and elements in the given array. Students will update the diagram with each statement. Guided practice and independent practice problems Exercise questions from Java Methods book	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Peer Feedback Project Rubric Lab work
8.1.12. AP.2	**Traversing Arrays**	Traverse the elements in a 1D array.	Do Now: MCQ practice using Arrays Live coding with teacher instruction on using iteration in traversing through elements of an array. Students participate in active discussion on trying different strategies. **Error analysis:** Students are given error ridden code segments with expected outputs. Students are asked to identify each error along with the fix to get the given output. Fortune Teller Lab activity. Exercise questions from Java Methods book RuneStone Academy practice questions		

8.1.12. AP.2	**Enhanced for Loop for Arrays**	Traverse the elements in a 1D array object using an enhanced for loop.	Do Now: Watch video and answer questions related to the video Live Coding with teacher instruction on introducing enhanced for loop. Taking an example from a previous lesson and rewriting it as an enhanced for loop. **Think-pair-share:** Students are given code segments that use both traditional and enhanced for loop statements. Students work in pairs and discuss their results. Unit 6 Progress check MCQ from AP Classroom Exercise questions from Java Methods book.		
8.1.12. AP.1 8.1.12. AP.5	**Developing Algorithms Using Arrays**	For algorithms in the context of a particular specification that requires the use of array traversals: a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Do Now: Review on using iteration in an array **Pair programming:** Students use pair programming to solve an array based free response question. One student is the driver for Part A while the other navigates and vice versa for Part B. They use scoring guidelines to grade their solution. Their different solutions are shared in class. Unit 6 Progress check FRQ from AP Classroom. RuneStone Academy practice questions from Unit 6.		

Unit # 7: ArrayList

Enduring Understandings:	Essential Questions:
● To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value. ● Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values. ● While programs are typically designed to achieve a specific purpose, they may have unintended consequences.	● Why is an ArrayList more appropriate for storing your music playlist, while an array might be more appropriate for storing your class schedule? ● How can we use statement execution counts to choose appropriate algorithms? ● What personal data is currently being collected, and how?

Interdisciplinary Connections

S-ID: Summarize, represent, and interpret data on a single count or measurement variable
Summarize, represent, and interpret data on two categorical and quantitative variables

Example: Students analyze and interpret data in an array or arraylist. Collection of temperatures in a given month or collections of grades of a student in a high school are some examples of data.

S-IC: Understand and evaluate random processes underlying statistical experiments
Make inferences and justify conclusions from sample surveys, experiments, and observational studies

Example: Students use the data to analyze different sorting and searching algorithms and their efficiency.

Career/Real World Connections

Example: List of transactions for credit cards.
List of students from a high school.

Guiding / Topical Questions with Specific Standards	Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies

8.1.12. AP.2	**Introduction to ArrayList**	Represent collections of related object reference data using ArrayList objects.	Do Now: Pros and Cons of array Introduction to ArrayList. Live coding with teacher instruction on introducing ArrayList. Guided practice and independent practice exercise questions RuneStone Academy on introduction to ArrayList	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Peer Feedback Project Rubric Lab work
8.1.12. AP.2	**ArrayList Methods**	Represent collections of related object reference data using ArrayList objects.	Do Now: Watch videos from AP classroom Introduction to ArrayList Methods Predict and compare: Students are given a FRQ from the previous unit. Students, in a small group, highlight the statements that need to be changed when using arraylist. Exercise questions from Java Methods book.		
8.1.12. AP.2	**Traversing ArrayLists**	For ArrayList objects: a. Traverse using a for or while loop b. Traverse using an enhanced for loop	Do Now: Review enhanced for loop Identify a subtask: As a group, students work on identifying subtasks in the given FRQ solution. Each student will take a subtask and work on implementing them. MCQ practice questions on traversing through ArrayList code segments. Practice MCQs from AP Classroom.		
8.1.12. AP.1 8.1.12. AP.5	**Developing Algorithms Using ArrayLists**	For algorithms in the context of a particular specification that requires the use of ArrayList traversals: a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Do Now: Review of ArrayList Methods **Predict and compare**: Students are asked to develop an algorithm using		

			ArrayList for the given FRQ that uses Array.		
8.1.12. AP.2 8.1.12. AP.8	**Searching**	Apply sequential/linear search algorithms to search for specific information in array or ArrayList objects.	Do Now: different searching algorithm Using iteration, students apply each searching algorithm using ArrayList. As a group, students discuss the efficiency of each searching algorithm and share their results with the class. Exercise questions using ArrayList searching algorithms.		
8.1.12. AP.2 8.1.12. AP.8	**Sorting**	Apply selection sort and insertion sort algorithms to sort the elements of array or ArrayList objects. Compute statement execution counts and informal run-time comparison of sorting algorithms.	Do Now: different sorting algorithm Using iteration, students apply each sorting algorithm using ArrayList. As a group, students discuss the efficiency of each sorting algorithm and share their results with the class. Exercise questions using ArrayList sorting algorithms. Unit 7 Progress Checks (MCQ / FRQ) from AP Classroom		
8.2.12. EC.3	**Ethical Issues Around Data Collection**	Explain the risks to privacy from collecting and storing personal data on computer systems.	Do Now: Share the risk of sharing personal details online. **Discussion group:** Students research risks to privacy from collecting and storing personal data on computer systems. Presentation: Students work on creating a presentation on what is privacy and how to protect your personal information online.		

			Play Google Developed game of InterLand.		

Unit # 8: 2D Array

Enduring Understandings:	**Essential Questions:**
● To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value. ● Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.	● Why might you want to use a 2D array to store the spaces on a game board or the pixels in a picture, rather than a 1D array or ArrayList? ● Why does the order in which elements are accessed in 2D array traversal matter in some situations?
Interdisciplinary Connections **N -VM: Perform operations on matrices and use matrices in applications.** **A -REI: Solve systems of equations** **Students are using matrices in the form of 2-dimensional arrays and developing games using Java.**	
Career/Real World Connections **Example: Game Development: Tic Tac Toe, Magic square building etc..**	

Guiding / Topical Questions with Specific Standards		**Content, Themes, Concepts, and Skills**	**Teaching Strategies**	**Instructional Resources and Materials**	**Assessment Strategies**
8.1.12. AP.2	**2D Arrays**	Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.	Do Now: Practice MCQ questions from College Board Introduction to 2D Arrays. Live coding with teacher instructions to syntax and uses of 2D arrays. Practice problems from the Java Methods book.	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Peer Feedback Project Rubric

8.1.12. AP.2	**Traversing 2D Arrays**	For 2D array objects: a. Traverse using nested for loops. b. Traverse using nested enhanced for loops. For algorithms in the context of a particular specification that requires the use of 2D array traversals: a. Identify standard algorithms. b. Modify standard algorithms. c. Develop an algorithm.	Do-Now: Watch 3 videos from AP classroom **Sharing and responding:** After completing a FRQ, share the answer with another student. Students exchange feedback, errors and fixes for the FRQ. Practice questions on how to traverse through 2D array questions. MCQ practice questions along with FRQ. Complete Unit 8 progress checks MCQ and FRQ.		

Unit # 9: Inheritance

Enduring Understandings: ● When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.	**Essential Questions:** ● How might the use of inheritance help in writing a program that simulates crops being grown in a virtual world? ● How does inheritance make programs more versatile?

Interdisciplinary Connections

Career/Real World Connections

Example: Polymorphism in real world example: ***Consider your Mobile phone. You can save your Contacts in it. Now suppose you want to save 2 numbers for one person. You can do it by saving the second number under the same name.***

Guiding / Topical Questions with Specific Standards		Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies
8.1.12. AP.5 8.1.12. AP.6	**Creating Superclasses and Subclasses**	Create an inheritance relationship from a subclass to the superclass.	Do Now Introduction to Inheritance. Live coding with teacher instruction on Super and subclasses. **Activating prior knowledge**: Students are given exercise questions to activate their prior knowledge. Worksheet on inheritance	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Lab work Peer Feedback Project Rubric
8.1.12. AP.5	**Writing Constructors for Subclasses**	Create an inheritance relationship from a subclass to the superclass.	Do Now		

8.1.12. AP.6			**Create a plan**: Students as a group create a plan with the existing super and subclasses and find the common attributes and behaviors among these classes. Exercise questions on subclasses		
8.1.12. AP.5 8.1.12. AP.6	**Overriding Methods**	Create an inheritance relationship from a subclass to the superclass.	Do Now (Review of super and subclasses) Identify Overriding methods on the given super and subclasses. Students are given flashcards with super class and subclass methods and asked them to categorize them.		
8.1.12. AP.5 8.1.12. AP.6	**super Keyword**	Create an inheritance relationship from a subclass to the superclass.	Do Now (Review of overriding methods) Live coding with teacher instruction on using a "super" keyboard! Independent practice on superclass and subclass.		
8.1.12. AP.5 8.1.12. AP.6	**Creating References Using Inheritance Hierarchies**	Define reference variables of a superclass to be assigned to an object of a subclass in the same hierarchy.	Do Now **Student response system** activity with students working in groups to identify each given statement would result in a compile time error or run time error. Lab work: With the superclasses and subclasses given to the students, they are asked to write test classes.		

8.1.12. AP.5 8.1.12. AP.6	**Polymorphism**	Call methods in an inheritance relationship.	Do Now Live coding with teacher instructions on Polymorphism. Exercise questions (worksheet) on Polymorphism.		
8.1.12. AP.5 8.1.12. AP.6	**Object Superclass**	Call Object class methods through inheritance.	Do Now Unit 9 Progress Check on MCQ/FRQ RuneStone Academy practice questions		

Unit # 10: Recursion

Enduring Understandings:	**Essential Questions:**
● Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values. ●	● What real-world processes do you follow that are recursive in nature? ● Why do programmers sometimes prefer using recursive solutions when sorting data in a large data set?

Interdisciplinary Connections

S-ID: Summarize, represent, and interpret data on a single count or measurement variable
Summarize, represent, and interpret data on two categorical and quantitative variables

Example: Students analyze and interpret data in an array or arraylist. Collection of temperatures in a given month or collections of grades of a student in a high school are some examples of data.

S-IC: Understand and evaluate random processes underlying statistical experiments
Make inferences and justify conclusions from sample surveys, experiments, and observational studies

Example: Students use the data to analyze different sorting and searching algorithms and their efficiency.

Career/Real World Connections

Example: File Systems in an operating system: finding files, deleting files, creating files and directories etc.
Library System: Sorting and searching for books in a library system.

Guiding / Topical Questions with Specific Standards	Content, Themes, Concepts, and Skills	Teaching Strategies	Instructional Resources and Materials	Assessment Strategies

8.1.8.AP.3:	**Recursion**	Determine the result of executing recursive methods.	Do Now (YouTube Video on Recursion) Introduction to Recursion. Complete worksheet for traversal of recursion code segments. Look for a pattern (Small group activity): Students are given a set of values for a recursion program and asked to look for a pattern to find the answer. RuneStone Academy assignment	Desktop with Eclipse (IDE) Java Methods Book RuneStone Academy Online Resource AP Classroom from CollegeBoard Website	Teacher Feedback from classwork Peer Feedback Project Rubric
8.1.12.AP.8	**Recursive Searching and Sorting**	Apply recursive search algorithms to information in String, 1D array, or ArrayList objects. Apply recursive algorithms to sort elements of array or ArrayList objects.	Padlet Activity on a different sorting and searching algorithm that uses recursion. Worksheet on traversal of sorting and searching algorithm. Code tracing: Students look at the recursive methods and determine how many times it executes. Progress Checks from AP Classroom.		

General Differentiated Instruction Strategies

- Leveled texts
- Chunking texts
- Choice board
- Socratic Seminar
- Tiered Instruction
- Small group instruction
- Guided Reading
- Sentence starters/frames
- Writing scaffolds
- Tangible items/pictures
- Adjust length of assignment
- Repeat, reword directions
- Brain breaks and movement breaks
- Brief and concrete directions
- Checklists for tasks
- Graphic organizers
- Assistive technology (spell check, voice to type)
- Study guides
- Tiered learning stations
- Tiered questioning
- Data-driven student partnerships
- Extra time

Possible Additional Strategies for Special Education Students, 504 Students, At-Risk Students, and English Language Learners (ELLs)

Time/General	Processing	Comprehension	Recall
● Extra time for assigned tasks ● Adjust length of assignment ● Timeline with due dates for reports and projects ● Communication system between home and school ● Provide lecture notes/outline	● Extra Response time ● Have students verbalize steps ● Repeat, clarify or reword directions ● Mini-breaks between tasks ● Provide a warning for transitions ● Reading partners	● Precise step-by-step directions ● Short manageable tasks ● Brief and concrete directions ● Provide immediate feedback ● Small group instruction ● Emphasize multi-sensory learning	● Teacher-made checklist ● Use visual graphic organizers ● Reference resources to promote independence ● Visual and verbal reminders ● Graphic organizers
Assistive Technology	**Assessments and Grading**	**Behavior/Attention**	**Organization**

● Computer/whiteboard ● Tape recorder ● Spell-checker ● Audio-taped books	● Extended time ● Study guides ● Shortened tests ● Read directions aloud	● Consistent daily structured routine ● Simple and clear classroom rules ● Frequent feedback	● Individual daily planner ● Display a written agenda ● Note-taking assistance ● Color code materials

Enrichment

The goal of Enrichment is to provide learners with the opportunity to participate in extension activities that are differentiated and enhance the curriculum. All enrichment decisions will be based upon individual student needs.

- Show a high degree of intellectual, creative and/or artistic ability and demonstrate this ability in multiple ways.
- Pose questions and exhibit sincere curiosity about principles and how things work.
- The ability to grasp concepts and make real world and cross-curricular connections.
- Generate theories and hypotheses and pursue methods of inquiry.
- Produce products that express insight, creativity, and excellence.
- Possess exceptional leadership skills.
- Evaluate vocabulary
- Elevate Text Complexity
- Inquiry based assignments and projects
- Independent student options
- Tiered/Multi-level activities
- Purposeful Learning Center
- Open-ended activities and projects
- Form and build on learning communities
- Providing pupils with experiences outside the ‘regular’ curriculum
- Altering the pace the student uses to cover regular curriculum in order to explore topics of interest in greater depth/breadth within their own grade level
- A higher quality of work than the norm for the given age group.
- The promotion of a higher level of thinking and makingconnections.
- The inclusion of additional subject areas and/or activities (cross-curricular).
- Using supplementary materials in addition to the normal range of resources.

English Language Learner (ELL) Resources

- Learning style quiz for students- http://www.educationplanner.org/students/self-assessments/learning-styles-quiz.shtml
- "Word clouds" from text that you provide-http://www.wordle.net/
- Bilingual website for students, parents and educators: http://www.colorincolorado.org/
- Learn a language for FREE-www.Duolingo.com
- Time on task for students-http://www.online-stopwatch.com/
- Differentiation activities for students based on their Lexile-www.Mobymax.com
- WIDA-http://www.wida.us/
- Everything ESL - http://www.everythingESL.net
- ELL Tool Box Suggestion Site http://www.wallwisher.com/wall/elltoolbox
- Hope4Education - http://www.hope4education.com
- Learning the Language http://blogs.edweek.org/edweek/learning-the-language/
- FLENJ (Foreign Language Educators of NJ) 'E-Verse' wiki: http://www.flenj.org/Publications/?page=135
- OELA - http://www.ed.gov/offices/OBEMLA
- New Jersey Department of Education- Bilingual Education information http://www.state.nj.us/education/bilingual/

Special Education Resources

- Animoto -Animoto provides tools for making videos by using animation to pull together a series of images and combining them with audio. Animoto videos or presentations are easy to publish and share. https://animoto.com
- Bookbuilder -Use this site to create, share, publish, and read digital books that engage and support diverse learners according to their individual needs, interests, and skills. http://bookbuilder.cast.org/
- CAST -CAST is a non-profit research and development organization dedicated to Universal Design for Learning (UDL). UDL research demonstrates that the challenge of diversity can and must be met by making curriculum flexible and responsive to learner differences. http://www.cast.org
- CoSketch -CoSketch is a multi-user online whiteboard designed to give you the ability to quickly visualize and share your ideas as images. http://www.cosketch.com/
- Crayon -The Crayon.net site offers an electronic template for students to create their own newspapers. The site allows you to bring multiple sources together, thus creating an individualized and customized newspaper. http://crayon.net/ Education Oasis -Education Oasis offers a collection of graphic organizers to help students organize and retain knowledge – cause and effect, character and story, compare and

contrast, and more! http://www.educationoasis.com/printables/graphic-organizers/

- Edutopia -A comprehensive website and online community that increases knowledge, sharing, and adoption of what works in K-12 education. We emphasize core strategies: project-based learning, comprehensive assessment, integrated studies, social and emotional learning, educational leadership and teacher development, and technology integration. http://www.edutopia.org/
- Glogster -Glogster allows you to create "interactive posters" to communicate ideas. Students can embed media links, sound, and video, and then share their posters with friends. http://edu.glogster.com/?ref=personal
- Interactives – Elements of a Story -This interactive breaks down the important elements of a story. Students go through the series of steps for constructing a story including: Setting, Characters, Sequence, Exposition, Conflict, Climax, and Resolution. http://www.learner.org/interactives/story/index.html
- National Writing Project (NWP) -Unique in breadth and scale, the NWP is a network of sites anchored at colleges and universities and serving teachers across disciplines and at all levels, from early childhood through university. We provide professional development, develop resources, generate research, and act on knowledge to improve the teaching of writing and learning in schools and communities. http://www.nwp.org
- Pacecar -Vocab Ahead offers videos that give an active demonstration of vocabulary with audio repeating the pronunciation, definition, various uses, and synonyms. Students can also go through flash cards which give a written definition and visual representation of the word. http://pacecar.missingmethod.com/