Unit 01: Primitive Types

Content Area: **Mathematics**
Course(s):
Time Period: **September**
Length: **7 blocks**
Status: **Awaiting Review**

Course Description & Instructional Notes

This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call preexisting methods to produce output. The use of preexisting methods for input is not prescribed in the course; however, input is a necessary part of any computer science course so teachers will need to determine how they will address this in their classrooms. Students will start to learn about three built-in data types and learn how to create variables, store values, and interact with those variables using basic operations. The ability to write expressions is essential to representing the variability of the real world in a program and will be used in all future units. Primitive data is one of two categories of variables covered in this course. The other category, reference data, will be covered in Unit 2.

Prior Knowledge:

Students should have completed the Introduction to Programming with Karel the Dog on CodeHS as their summer assignment.

Introduction to Programming in Java with Karel is meant to be an introductory unit for students that are fairly new to programming, or as a summer exercise for students to complete before the start of the school year. The concepts taught will help students with the units that follow, but it's important to note that the Karel unit does not follow the College Boards expected unit progress.

Instructional Notes:

- **Building Computational Thinking Practices**
  - When writing programs, programmers use mathematical expressions to represent the relationships between quantities in the real world. This requires programmers to apply the meaning of specific Java operators to these formulas and expressions. While practicing writing these expressions, students should begin to understand that programs are composed of a series of statements that use arithmetic operators. During the early stages of learning to program, have students look at existing program code, and ask them to explain what it does rather than having them develop program code from scratch. When writing code, errors are inevitable. To learn how to identify and correct errors, students need exposure to the error messages generated by the compiler. The ability to communicate to collaborative partners why a code segment will not compile or work as intended will aid students in being able to correct the error and build working programs that accomplish specific tasks.

- **Preparing for the AP Exam**
  - This unit provides a lot of the foundational content and skills that students will continue to draw on throughout the course. While much of what is covered in this unit is not explicitly

Unit 02: Using Objects

Content Area: **Mathematics**
Course(s):
Time Period: **October**
Length: **7**
Status: **Awaiting Review**

Course Description & Instructional Notes

In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to Math class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the String class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course but will be very important in Units 5 and 9 when teaching students how to write their own classes and about inheritance relationships.

Prior Knowledge:

Skills and topics from previous unit(s)

Instructional Notes:

- **Building Computational Thinking Practices**
    - The study of computer science involves implementing the design or specification for a program. This is the fun and rewarding part of computer science, because it involves putting a plan into practice to create a runnable program. In addition to developing their own programs, students should practice completing partially written program code to fulfill a specification. This builds their confidence and provides them opportunities to be successful during these early stages of learning. Programmers often rely on existing program code to build new programs. Using existing code saves time, because it has already been tested. By using the String class, students will learn how to interact with and utilize any existing Java class to create objects and call methods.

- **Preparing for the AP Exam**
    - During the free-response portion of the exam, students will be required to call methods of classes that they haven't been exposed to prior to the exam. Students should get plenty of practice identifying the proper parameters to use when calling methods of classes that are provided to them. Often, students struggle with free-response questions that require them to work with the String class. Using the Java Quick Reference (p. 209) regularly during class will help students become more familiar with this resource prior to the exam. Paying close attention to the method descriptions will ensure that students use the correct type and order of parameters when calling String methods. Practice close reading techniques with students prior to the exam, such as underlining keywords, return types, and parameters. Students have approximately 20 minutes to read, process, and answer each of the four free-response questions. These close reading techniques are valuable in helping students process the questions quickly without

inadvertently missing key information.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

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 solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

Essential Questions

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?

Student Learning Objectives

Explain the relationship between a class and an object

Create classes with instance variables

Create and use constructors

Create objects by calling constructors with parameters

Explain the purpose of method overloading

Create classes that overload methods

Explain how null objects do not point to any particular object data

Create programs that use other classes as a client to solve a specific problem

Create and call class methods

Call non-static void methods without parameters

Call non-static void methods with parameters

Create and call non-void methods with parameters and return values

Create String objects for a String class

Concatenate Strings using operators and escape sequences

Create Integer and Double objects for wrapper classes

Call Integer and Double methods for wrapper classes

Call static methods

Evaluate expressions that use the Math class methods

Vocabulary & Learning Experiences

Essential Academic Vocabulary :

class - defines a new data type. It is the formal implementation, or blueprint, of the attributes and behaviors of the objects of that class.

object - a specific instance of a class with defined attributes. Objects are declared as variables of a class type.

constructors - code that is used to create new objects and initialize the object's attributes.

new - keyword used to create objects with a call to one of the class's constructors.

instance variables - define the attributes for objects.

methods - define the behaviors or functions for objects.

dot (.) operator - used to access an object's methods.

parameters (arguments) - the values or data passed to an object's method inside the parentheses in the method call to help the method do its job.

return values - values returned by methods to the calling method.

immutable - String methods do not change the String object. Any method that seems to change a string actually creates a new string.

wrapper classes - classes that create objects from primitive types, for example the Integer class and Double class.

new is used to create a new object.

null is used to indicate that an object reference doesn't refer to any object yet.

The following String methods and constructors, including what they do and when they are used, are part of the Java Quick Reference in the AP exam:

String(String str) : Constructs a new String object that represents the same sequence of characters as str.

int length() : returns the number of characters in a String object.

String substring(int from, int to) : returns the substring beginning at index from and ending at index (to – 1).

String substring(int from) : returns substring(from, length()). A string identical to the single element substring at position index can be created by calling substring(index, index + 1

int indexOf(String str) : returns the index of the first occurrence of str; returns -1 if not found.

boolean equals(String other) : returns true if this (the calling object) is equal to other; returns false otherwise.

int compareTo(String other) : returns a value < 0 if this is less than other; returns zero if this is equal to other; returns a value > 0 if this is greater than other.

The following Integer methods and constructors, including what they do and when they are used, are part of the Java Quick Reference.

Integer(value): Constructs a new Integer object that represents the specified int value.

Integer.MIN_VALUE : The minimum value represented by an int or Integer.

Integer.MAX_VALUE : The maximum value represented by an int or Integer.

int intValue() : Returns the value of this Integer as an int.

The following Double methods and constructors, including what they do and when they are used, are part of the Java Quick Reference Guide given during the exam:

Double(double value) : Constructs a new Double object that represents the specified double value.

double doubleValue() : Returns the value of this Double as a double.

The following static Math methods are part of the Java Quick Reference:

int abs(int) : Returns the absolute value of an int value (which means no negatives).

double abs(double) : Returns the absolute value of a double value.

double pow(double, double) : Returns the value of the first parameter raised to the power of the second parameter.

double sqrt(double) : Returns the positive square root of a double value.

double random() : Returns a double value greater than or equal to 0.0 and less than 1.0 (not including 1.0)!

(int)(Math.random()*range) + min moves the random number into a range starting from a minimum number. The range is the (max number - min number + 1). For example, to get a number in the range of 5 to 10, use the range 10-5+1 = 6 and the min number 5: (int)(Math.random()*6) + 5).

Planned Learning Experiences

- Using manipulatives: When introducing students to the idea of creating objects, you can use a cookie cutter and modeling clay or dough, with the cutter representing the class and cut dough representing the objects. For each object cut, write the instantiation. Ask students to describe what the code is doing and how the different parameter values(e.g. thickness, color) change the object that was created.

- Marking the text: Provide students with several statements that define a varible and create an object on a single line. Have studnts mark up the statements by circling the assignment operator and the *new* keyword. Then, have students underline the variable type and the constructor. Lastly, have them draw a rectangle around the list of actual parameters beiong passed to the constructor. Using these marked-up statements, ask students to create several new variables and objects.

- Think-pair-share: Provide students with several code segments, each with a missing expression that would contain a call to a method in the Math class, and a description of the intended outcome of each code segment. Ask them which statement should be used to complete the code segment. Have them share their responses with a partner to compare answers and come to agreeement, and then have groups share with the entire class.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students do a deep dive into java packages and documentation for classes as a client
- Have students research and investigate a floating point error.
- Have students research and use other String methods. Have them display the use of the methods for the class.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference
- Scaffolding or template with class, object skeletons

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference
- Scaffolding or template with class, object skeletons

Unit 03: Boolean Expressions and if Statements

Content Area: **Mathematics**
Course(s):
Time Period: **October**
Length: **6 blocks**
Status: **Awaiting Review**

Course Description & Instructional Notes

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators. The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

Prior Knowledge:

Skills and topics from previous unit(s)

Instructional Notes:

- **Building Computational Thinking Practices**
  - Selection allows programmers to incorporate choice into their programs: to create games that react to interactions of the user, to develop simulations that are more real world by allowing for variability, or to discover new knowledge in a sea of information by filtering out irrelevant data. Students should be able to write program code that uses conditional statements for selection. Have students write their program code on paper and trace it with different sample inputs before writing code on the computer. Programmers need to make design decisions when creating programs that determine how a program specification will be implemented. Typically, there are many ways in which statements can be written to yield the same result, and this final determination is dictated by the programmer. Exposing students to many different code segments that yield equivalent results allows them to be more confident in their solution and helps expose them to new ways of solving the problem that may be better than their current solution.

- **Preparing for the AP Exam**
  - The study of computer science involves the analysis of program code. On the multiplechoice section of the exam, students will be asked to determine the result of a given program code segment based on specific input and on the behavior of program code in general. Students should be able to determine the result of program code that uses conditional statements and nested conditional statements to represent nonlinear processes in a program. Often, students will write program code that mishandles one of the given conditions. The ability to trace program code can be valuable when testing programs to ensure that all conditions are met. Testing for the different expected behaviors of conditional statements is a critical part of program development and is useful when writing program code or analyzing given code

segments. Students should develop sample test cases to illustrate each unique behavior to aid in finding errors and validating results.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

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.

Essential Questions

Hopw can you use different conditional statements to write a pick - your- own path intercative story?

Why is selection a necessary part of programming?

Student Learning Objectives

Create and call objects and methods

Boolean expressions and relational operators

Represent branching logical processes by using flowcharts

Use if statements

Use if/else statements

Use an else if statement

Evaluate compound Boolean expressions in program code

Compare and contrast equivalent Boolean expressions

Apply De Morgan's Laws to Boolean expressions

Compare object references using Boolean expressions in program code

Vocabulary & Learning Experiences

Essential Academic Vocabulary

Block of statements - One or more statements enclosed in an open curly brace '{' and a closing curly brace '}'.

Boolean expression - A mathematical or logical expression that is either true or false.

complex conditional - A Boolean expression with two or more conditions joined by a logical and '&&' or a logical or '||'.

conditional - Used to execute code only if a Boolean expression is true.

DeMorgan's Laws - Rules about how to distribute a negation on a complex conditional.

logical and - Used to only execute the following statement or block of statements if both conditions are true

logical or - Used to execute the following statement or block of statements if one of the conditions are true

negation - turns a true statement false and a false statement true

short circuit evaluation - The type of evaluation used for logical and '&&' and logical or '||' expressions. If the first condition is false in a complex conditional with a logical and the second condition won't be evaluated. If the first condition is true is a complex conditional with a logical or the second condition won't be evaluated.

if (Boolean expression) - used to start a conditional statement. This is followed by a statement or a block of statements that will be executed if the Boolean expression is true.

else - used to execute a statement or block of statements if the Boolean expression on the if part was false.

else if (Boolean expression) - used to have 3 or more possible outcomes such as if x is equal, x is greater than, or x is less than some value. It will only execute if the condition in the 'if' was false and the condition in the else if is true.

Planned Learning Experiences

- Code tracing**:** Provide students with several code segments that contain conditional statements. Have students trace various sample inputs. keeping track of the statements that get executed and the order in which they are executed. This can help students find errors and validate results.

- Pair programming: Have students work with a putner to create a "guess checker" that could be used as part of a Larger gane. Students compare four given digits to a preexisting four-digit code that is stored in individual values. Their program should provide output containing the number of correct digits in correct locations. as well as the number of correct digits in incorrect locations_. This program can be continually improved as students leam about nested conditional statements and compound Boolean exression

- Diagramming**:** Have students create truth tables by listing all the possible true and false combinations and corresponding Boolean values for a given compound Boolean expression. Once students have created the truth table. provide students with input values. Have students determine the value of each individual Boolean expression and use the truth table to determine the result of the compound Boolean expression.

- Student response system: Provide students with a code segment that utilizes conditional statements and a compound Boolean expression and ask them to choose an equivalent code segment that uses a nested conditional statement (and vice versa). Have them report their responses using a student response system.

- Predict and compare: Have students predict the output of several different code segments that compare object references—some that use . equals ( ) and some that use Once done. have them create a program that contains those code segments and compue the actual and expected results. This is best illustrated using a simple class that you write yourself.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

[NJSLS Standards - Mathematics]

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them;

and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7×8 equals the well remembered $7 \times 5 + 7 \times 3$, in preparation for learning about the distributive property. In the expression $x2 + 9x + 14$, older students can see the 14 as 2×7 and the 9 as $2 + 7$. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see $5 – 3(x – y) 2$ as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation $(y – 2)/(x – 1) = 3$. Noticing the regularity in the way terms cancel when expanding $(x – 1)(x + 1)$, $(x – 1)(x2 + x + 1)$, and $(x – 1)(x3 + x2 + x + 1)$ might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.

MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students trace truth tables for more complex compound boolean expressions.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 04: Iteration

Content Area: **Mathematics**
Course(s):
Time Period: **November**
Length: **7**
Status: **Awaiting Review**

Course Description & Instructional Notes

This unit focuses on iteration using while and for loops. As you saw in Unit 3, Boolean expressions are useful when a program needs to perform different operations under different conditions. Boolean expressions are also one of the main components in iteration. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations. Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays. In addition, it is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units.

Prior Knowledge:

skills and topics from previous units

Instructional Notes:

- **Building Computational Thinking Practices**
  - Students should be able to determine the result of program code that uses iterative statements to represent nonlinear processes in a program. Students should practice determining the number of times a given loop structure will execute. Spending time analyzing existing program code provides opportunities for students to better understand how iterative structures can be set up and used to solve their own problems, as well as the implications associated with code changes, such as how using a different iterative structure might change the result of a set of program code. Students should be able to implement program code that uses iterative statements to represent nonlinear processes. Understanding how to write program code that repeats allows students to write programs to solve a wider variety of problems, including those that use data, which will be covered in Units 6, 7, and 8.

- **Preparing for the AP Exam**
  - While the concept of iteration is tested in isolation on the multiple-choice exam, it is a foundational concept that students will use along with other topics, such as data structures, on free-response questions. Often, students struggle in situations that warrant variation in the Boolean condition of loops, such as when they want to terminate a loop early. Early termination of a loop requires the use of conditional statements within the body of the loop. If the order of the program statements is incorrect, the early termination may be triggered too early or not at all. Provide students with practice ordering statements by giving them strips of paper, each with a line of program code that can be used to assemble the correct and incorrect solutions. Ask them to reassemble the code and trace it to see if it is correct. Using manipulatives in this way makes it easier for students to rearrange the order of the program code to determine if it is in

the correct order.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

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.

Essential Questions

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?

Student Learning Objectives

Represent iterative processes using a while loop

Execute a return or break statement inside an iteration statement to halt the loop and exit the method or constructor

Develop an algorithm

Identify if an integer is or is not evenly divisible by another integer

Determine a minimum or maximum value

Compute a sum, average, or mode

Represent iterative processes using a for loop

Develop an algorithm using Strings

Find if one or more substrings has a particular property

Determine the number of substrings that meet specific criteria

Create a new string with the characters reversed

Represent nested iterative processes

Compute statement execution counts and informal run-time comparison of iterative statements.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

Body of a Loop - The single statement or a block of statements that can be repeated (a loop may not execute at all if the condition is false to start with). In Java the body of the loop is either the first statement following a while or for loop is the body of the loop or a block of statements enclosed in { and }.

For Loop - A loop that has a header with 3 optional parts: initialization, condition, and change. It does the initialization one time before the body of the loop executes, executes the body of the loop if the condition is true, and executes the change after the body of the loop executes before checking the condition again.

Infinite Loop - A loop that never ends.

Loop - A way to repeat one or more statements in a program.

Nested Loop - One loop inside of another.

Out of Bounds error - A run-time error that occurs when you try to access past the end of a string or list in a loop.

Trace Code - Writing down the values of the variables and how they change each time the body of the loop executes.

While Loop - A loop that repeats while a Boolean expression is true.

while - used to start a while loop

for - used to start a for loop or a for each loop

System.out.println(variable) - used to print the value of the variable. This is useful in tracing the execution of code and when debugging.

Planned Learning Experiences

Jigsaw As a whole class. look at a code segment containing iteration and the resulting output. Afterward. divide students into groups. and provide each group with a slightly modified code segment. After groups have determined how the result changes based on their modified segment. have them get together with students who investigated a different version of the code seament and share their conclusions.

Note-taking Provide students 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. Have them annotate what each statement does in the method. Then. ask students to 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

Simplify the problem Provide students with several code segments containing teration_ For each segment have students trace through the execution of a loop with smaller bounds to see what boundary cases are considered and then use that information to determine the

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students find patterns in nested loop iteration counts.

Modification: Special Education

- Pair programming with another student

- Print out slides for students to reference

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 05: Writing Classes

Content Area: **Mathematics**
Course(s):
Time Period: **December**
Length: **7 blocks**
Status: **Awaiting Review**

Course Description & Instructional Notes

This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes. The ability to accurately model real-world entities in a computer program is a large part of what makes computer science so powerful. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. Students will build on what they learn in this unit to represent relationships between classes through hierarchies, which appear in Unit 9. The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit.

Prior Knowledge:

skills and topics from previous units

Instructional Notes:

- **Building Computational Thinking Practices**
    - Computer scientists use computers to study and model the world, and this requires designing program code to fit a given scenario. Spending time to plan out the program up front will shorten overall program development time. Practicing elements of the iterative development process early and often will help create more robust programs and avoid logical errors. Because revisions to program code are rarely completed by the original author, documenting program code is very important. It allows programmers to more quickly understand how a code segment functions without having to analyze the program code itself. Well-written documentation includes a description of the behavior, as well as the initial conditions that must be met for a code segment to work as intended. Sometimes students struggle to explain why a code segment will not compile or work as intended. It helps to have students work with a collaborative partner to practice verbally explaining the issues. The ability to explain why a code segment isn't working correctly assists in the development of solutions.

- **Preparing for the AP Exam**
    - On the free-response section of the exam, students are required to design program code that demonstrates the functionality available for an object of a new class, based on the prompt's specification. This process includes identifying the attributes (data) that define an object of a class and the behaviors (methods) that define what an object of a class can do. Students should have many opportunities in this course to design their own classes based on program specifications and on observations of real-world objects. Once the new class is designed, students should be able to implement program code to create a new type by creating a class. The behaviors of an object are expressed through writing methods in the class, which include expressions, conditional statements, and iterative statements. By being able to create their own

classes, programmers are not limited to the existing classes provided within the Java libraries and can therefore represent their own ideas through classes.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

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 apply to the subclasses.

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.

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different output values.

While programs are typically designed to acheive a specific purpose, they may have unintended consequences.

Essential Questions

How can using a model 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?

Student Learning Objectives

Students will be able to...

- Designate access and visibility constraints to classes, data, constructors, and methods.
- Designate private visibility of instance variables to encapsulate the attributes of an object.
- Define instance variables for the attributes to be initialized through the constructors of a class.
- Describe the functionality and use of program code through comments.
- Define behaviors of an object through non-void methods without parameters written in a class.
- Define behaviors of an object through void methods with or without parameters written in a class.

- Define behaviors of an object through non-void methods with parameters written in a class.
- Define behaviors of a class through static methods.
- Define the static variables that belong to the class.
- Explain where variables can be used in the program code.
- Evaluate object reference expressions that use the keyword this.
- Explain the ethical and social implications of computing systems.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

Class - A class defines a type and is used to define what all objects of that class know and can do.

Compiler - Software that translates the Java source code (ends in .java) into the Java class file (ends in .class).

Compile time error - An error that is found during the compilation. These are also called syntax errors.

Constructor - Used to initialize fields in a newly created object.

Field - A field holds data or a property - what an object knows or keeps track of.

Java - A programming language that you can use to tell a computer what to do.

Main Method - Where execution starts in a Java program.

Method - Defines behavior - what an object can do.

Object - Objects do the actual work in an object-oriented program.

Syntax Error - A syntax error is an error in the specification of the program.

class - used to define a new class

public - a visibility keyword which is used to control the classes that have access. The keyword public means the code in any class has direct access.

private - a visibility keyword which is used to control the classes that have access. The keyword private means that only the code in the current class has direct access.

Planned Learning Experiences

Kinesthetic learning: Have students break into groups of 4–5 to play board games. Ask them to play the game for about 10 minutes. While they play the game, they should keep track of the various nouns they encounter and actions that happen as part of the game. The nouns can be represented in the computer as classes, and the actions are the behaviors. At the end of game play, ask students to create UML diagrams for the identified classes.

Marking the text: Present students with specifications, and have them highlight or underline any preconditions (both implicit and explicit) that exist for the method to function. This includes information about parameters, such as object references not being null.

Create a plan: When asked to write a method, have students write an outline using pseudocode with paper and pencil. Then, 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 beginning to program a solution on the computer.

Paraphrase: Provide students with several example classes that utilize static variables for unique identification numbers or for counting the number of objects that have been created, but do not provide any description or documentation for the code. Have students spend time creating objects and calling the static methods to investigate how the static variables behave, then have them document the code appropriately to describe how each class utilizes static variables and methods.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students create multiple classes that interact with each other.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 06: Arrays

Content Area: **Mathematics**
Course(s):
Time Period: **January**
Length: **3 blocks**
Status: **Awaiting Review**

Course Description & Instructional Notes

This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. Using a data structure along with iterative statements with appropriate bounds will allow for similar treatment to be applied more easily to all values in the collection. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. In this unit, we apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well. Additional standard algorithms, such as standard searching and sorting algorithms, will be covered in the next unit.

Prior Knowledge:

skills and topics from previous units

Instructional Notes:

- **Building Computational Thinking Practices**
  - Students should be able to implement program code to create, traverse, and manipulate elements in a 1D array. Traversing elements of a 1D array can be accomplished in multiple ways. Programmers need to make decisions about which loop structure is most effective given the problem they are trying to solve. Some loop structures, such as the enhanced for loop, only allow programmers to examine the data stored in a 1D array structure, while other loop structures allow the data to be manipulated. Students should be able to identify and correct errors related to traversing and manipulating 1D array structures. A common run-time error that programmers experience is an out-of-bounds error, which occurs when the program tries to access an element that is beyond the range of elements in a collection. Students should double-check the values of the index being used on at least the initial and final loop iterations to ensure that they aren't out of bounds.

- **Preparing for the AP Exam**
  - A specific iterative structure is commonly used to traverse an array: starting at the beginning and moving toward the last element in the array. When students are asked to determine the result of program code that contains arrays, they often use this same iterative structure. Knowing this iterative structure can be helpful when students use tracing to determine what an algorithm is doing. Students can follow this traversal for the first few iterations and apply that pattern to the remaining elements in the array. When preparing for the free-response questions, students should be come familiar with how existing algorithms work, rather than just memorizing the program code. Have students write out an algorithm on paper and test it using manipulatives. This allows students to experience the algorithm on a deeper level than if they simply program it. A strong understanding of how existing algorithms work allows

programmers to make modifications to those algorithms to accomplish similar tasks.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

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 creatingindividual 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.

Essential Questions

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?

Student Learning Objectives

Students will be able to...

Traverse the elements in a 1D Array.

Traverse the elements in a 1D array object using an enhanced for loop.

For algorithms in the context of a particular specification that requires the use of array traversals: identify standard algorithms, modify standard algorithms, develop an algorithm.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

Array - An array can hold many items (elements) of the same type. You can access an item (element) at an index and set an item (element) at an index.

Array Declaration - To declare an array specify the type of elements that will be stored in the array, then ([]) to show that it is an array of that type, then at least one space, and then a name for the array. Examples: int[] highScores; String[] names;

Array Creation - To create an array type the name and an equals sign then use the new keyword, followed by a space, then the type, and then in square brackets the size of the array (the number of elements it can hold). Example: names = new String[5];

Array Index - You can access and set values in an array using an index. The first element in an array called arr is at index 0 arr[0]. The last element in an array is at the length minus one - arr[arr.length - 1].

Array Initialization - You can also initialize (set) the values in the array when you create it. In this case you don't need to specify the size of the array, it will be determined from the number of values that you specify. Example: int[] highScores = {99,98,98,88,68};

Array Length - The length of an array is the number of elements it can hold. Use the public length field to get the length of the array. Example: given int[] scores = {1,2,2,1,3,1};, scores.length equals 6.

Element Reference - A specific element can be referenced by using the name of the array and the element's index in square brackets. Example: scores[3] will return the 4th element (since index starts at 0, not 1). To reference the last element in an array, use array[array.length - 1]

For-each Loop - Used to loop through all elements of an array. Each time through the loop the loop variable will be the next element in the array starting with the element at index 0, then index 1, then index 2, etc.

Out of Bounds Exception - An error that means that you tried to access an element of the array that doesn't exist maybe by doing arr[arr.length]. The first valid indices is 0 and the last is the length minus one.

for - starts both a general for loop and a for-each loop. The syntax for a for each loop is for (type variable : array). Each time through the loop the variable will take on the next value in the array. The first time through the loop it will hold the value at index 0, then the value at index 1, then the value at index 2, etc.

static - used to create a class method, which is a method that can be called using the class name like Math.abs(-3).

Planned Learning Experiences

Diagramming: Provide students with several prompts to create and access elements in an array. After they have determined the code for each prompt, have students draw a memory diagram that shows references and the arrays they point to. Have students update the diagram with each statement to demonstrate how changing the contents through one array reference effects all the array references for this array.

Error analysis: Provide students with several error-ridden code segments containing array traversals along with the expected output of each segment. Ask them to identify any errors that they see on paper and to suggest fixes to provide the expected output. Have them type up their solutions in an IDE to verify their work.

Think-pair-share: Ask students to consider two program code segments that are meant to yield the same result: one using a traditional for loop and one using a for each loop. Have them take a few minutes to think independently about whether the two segments accomplish the same result and, if not, what changes could be made in order for that to happen. Then, ask students to work with their partners to come up with situations where it would make sense to use one type of loop over the other before sharing with the whole class.

Pair programming: Have students use pair programming to solve an array-based free-response question. Have one student be the driver for Part A while the other navigates, then have them switch for Part B. Once they are done, have partners switch solutions with another group and work through the scoring guidelines to "grade" that solution. Spend time as a class discussing the different approaches students used.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students attempt to code FRQs from previous AP Exams.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference
- Allow use of calculator

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 07: ArrayList

Content Area: **Mathematics**
Course(s):
Time Period: **January**
Length: **6 blocks**
Status: **Awaiting Review**

Course Description & Instructional Notes

As students learned in Unit 6, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set. In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Prior Knowledge:

Instructional Notes:

- **Building Computational Thinking Practices**
  - Students need to consider the impact using ArrayList rather than an array has on the structure of their program code. This includes considering the use of ArrayList methods and the flexibility of a structure with a dynamic size. For instance, the use of an ArrayList will require students to analyze program code that uses method calls. Providing students with practice writing programs for data sets of undetermined sized—or at least larger than they would be able to analyze easily by hand—presents a more relevant and realistic experience with data. Additionally, this requires students to focus more on the algorithm and ensuring that it will work in all situations rather than on an individual result. With larger data sets, programmers become concerned with the amount of time it will take for their program code to run. Students should have practice determining the number of times a code segment executes; this can help them gain an idea of how long it will take to run a program on a data set of a given size.
  -
- **Preparing for the AP Exam**
  - When writing solutions to free-response questions that involve the use of an ArrayList, students are often asked to insert or delete elements from the ArrayList. In these cases, adjustments will need to be made to the loop counter to account for skipping an element or attempting to access elements that no longer exist. Students may also be asked to traverse multiple data structures simultaneously. These data structures may be a mixture of array and ArrayList objects. As such, it is very easy for students to become confused about how elements are accessed and manipulated within each structure. Additionally, the size of the data structures may not be the same. In these situations, itis best to have different index variables and bounds checks for each structure to avoid accessing elements that are out-of-bounds.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

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 individuals 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 input values.

While programs are typically designed to acheive a specific purpose, they may have unintended consequences.

Essential Questions

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?

Student Learning Objectives

Students will be able to...

Represent collections of related object reference data using ArrayList objects.

For ArrayList objects:traverse using a for or while loop, traverse using an enhanced for loop.

For algorithms in the context of a particular specification that requires the use of ArrayList traversals: Identify standard algorithms, Modify standard algorithms, and Develop an algorithm.

Apply sequential/linear search algorithms to search for specific information in array or ArrayList objects.

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.

Explain the risks to privacy from collecting and storing personal data on computer systems.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

Autoboxing - Automatically wrapping a primitive type in a wrapper class object. For instance if you try to add an int value to a list, it will automatically be converted to an Integer object.

Abstract Method - A method that only has a declaration and no method body (no code inside the method).

ArrayList - An ArrayList can hold many objects of the same type. It can grow or shrink as needed. You can add and remove items at any index.

Add - You can add an object to the end of a list using listName.add(obj). You can add an object at an index of a list using add(index,obj). This will first move any objects at that index or higher to the right one position to make room for the new object.

Declaration - To declare an ArrayList use ArrayList<Type> name, where Type is the class name for the type of objects in the list. If you leave off the <Type> it will default to Object.

Creation - To create an ArrayList use new ArrayList<Type>, where Type is the class name for the type of objects you want to store in the list. There are other classes that implement the List interface, but you only need to know the ArrayList class for the exam.

Get - To get an object at an index from a list use listName.get(index).

Index - You can access and set values in a list using an index. The first element in a list called list1 is at index 0 list1.get(0). The last element in a list is at the length minus one - list1[list1.size() - 1].

Remove - To remove the object at an index use ListName.remove(index). This will move all object past that index to the left one index.

Set - To set the value at an index in a list use listName.set(index,obj).

Size - Use listName.size() to get the number of objects in the list.

Wrapper Class - Classes used to create objects that hold primitive type values like Integer for int, Double for double and Boolean for boolean.

Unboxing - Automatically converting a wrapper object like an Integer into a primitive type such as an int.

Planned Learning Experiences

<u>Predict and compare</u> Have students look at the code they wrote to solve the free-response question in Unit 6 (or other code from Unit 6) on paper, and have them rewrite it using an ArrayList. Have them highlight the parts that need to be changed and determine how to change them. Then, have students type up the changes in an IDE and confirm that the program still works as expected.

<u>Identify a subtask</u> Have students read through an ArrayList-based free-response question in groups, and have them identify all subtasks. These subtasks could be conditional statements, iteration, or even other methods.

Once the subtasks have been identified, divide the subtasks among the group members, and have students implement their given subtask. When all students are finished, have them combine the subtasks into a single solution.

Discussion group Discuss the algorithm necessary to search for the smallest value in an ArrayList. Without explaining what you are doing, change the Boolean expression so that it will find the largest value, and ask students to describe what the resulting algorithm will do. Then, change the algorithm to store and return the location of the largest value, and discuss the change.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students brainstorm real life applications of ArrayLists.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference

- Allow use of calculator

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 08: 2D Array

Content Area: **Mathematics**
Course(s):
Time Period: **February**
Length: **6**
Status: **Awaiting Review**

Course Description & Instructional Notes

In Unit 6, students learned how 1D arrays store large amounts of related data. These same concepts will be implemented with two-dimensional (2D) arrays in this unit. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplished this is in row-major order, but it is important to cover additional traversal patterns, such as back and forth or column-major.

Prior Knowledge:

Instructional Notes:

- **Building Computational Thinking Practices**
  - Students should be able to determine the result of program code that traverses and manipulates the elements in a 2D array. Traversals of 2D arrays typically require a set of nested loops. Often the extra dimension of a 2D array is difficult for students to envision. Providing students with practice analyzing and tracing traversals of a 2D array, as well as providing them partial program code to complete, helps students take this more abstract concept and make it concrete and replicable. Because 2D arrays are traversed using nested loops, the number of times a code segment executes is multiplied. In a nested loop, the inner loop must complete all iterations before the outer loop can continue. It helps to provide students with sample code that will print the values in a 2D array. Teachers can an IDE that shows access to 2D arrays visually and keeps track of the execution count.

- **Preparing for the AP Exam**
  - The free-response portion of the exam always includes one question that requires students to write program code involving 2D arrays. Because 2D arrays are arrays where each element is an array, it is not uncommon for the question to require students to write solutions involving array or ArrayList objects as well. While there is a specific nested structure to traverse elements in a 2D array in rowmajor order, this structure can be modified to traverse 2D arrays in other ways, such as column-major, by switching the nested iterative statements. Additional modifications can be made to traverse rows or columns in different ways, such as back and forth or up and down. However, when making these adjustments, students often neglect to adjust the bounds of the iterative statements appropriately. Students should practice traversing 2D arrays in these nonstandard ways, being sure to test the boundary conditions of the iterative statements, to be prepared for this type of freeresponse question.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

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.

Essential Questions

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?

Student Learning Objectives

Students will be able to...

Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.

For 2D array objects: Traverse using nested for loops and Traverse using nested enhanced for loops.

For algorithms in the context of a particular specification that requires the use of 2D array traversals: Identify standard algorithms, Modify standard algorithms, and Develop an algorithm.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

2d Array - An array that holds items in a two dimensional grid. You can think of it as storing items in rows and columns (like a bingo card or battleship game). You can access an item (element) at a given row and column index.

2d Array Declaration - To declare an array, specify the type of elements that will be stored in the array, then ([][]) to show that it is a 2d array of that type, then at least one space, and then a name for the array. Examples: int[][] seats; String[][] seatingChart;

2d Array Creation - To create a 2d array, type the name and an equals sign then use the new keyword, followed by a space, then the type, and then [numRows][numCols]. Example: seatingChart = new String[5][4];. This will have 5 rows and 4 columns.

2d Array Index - You can access and set values in a 2d array using the row and column index. The first element in an array called arr is at row 0 and column 0 arr[0][0].

2d Array Initialization - You can also initialize (set) the values in the array when you first create it. In this case you don't need to specify the size of the array, it will be determined from the number of values that you specify. Example: String[][] seatingInfo = { {"Jamal", "Maria"}, {"Jake", "Suzy"}, {"Emma", "Luke"}}; This will create a 2d array with 3 rows and 2 columns.

2d Array Number of Rows - The number of rows (or height) is the length of the outer array. For an array arr use arr.length to get the number of rows in the array.

2d Array Number of Columns - The number of columns (or width) is the length of the inner array. For an array arr use arr[0].length to get the number of columns.

nested for loop - A for loop inside of another for loop. These are used to loop through all the elements in a 2d array. One loop can work through the rows and the other the columns.

out of bounds error - This happens when a loop goes beyond the last valid index in an array. Remember that the last valid row index is arr.length - 1. The last valid column index is arr[0].length - 1.

Planned Learning Experiences

<u>Using manipulatives</u> Use different-sized egg cartons or ice cube trays with random compartments filled with small toys or candy. Create laminated cards with the code for the construction of, and access to, a 2D array, leaving blanks for the name and size dimensions. Have students fill in the missing code that would be used to represent the physical 2D array objects and access the randomly stored elements.

<u>Activating prior knowledge</u> When first introducing 2D arrays and row-major traversal, ask students which part of the nested for loop structure loops over a 1D array. Based on what they know about the traversal of 1 D array structures, ask them to calculate the number of times the inner loop executes.

<u>Sharing and responding</u> As a class, create a set of test cases to be used with answers to a free-response question. Have students write their answers to the free-response question individually on paper. After exchanging solutions with another student, ask students to find errors or validate results of their peers' code by tracing the code with the developed test cases. Allow students an opportunity to provide feedback on the program code as well as the results of each test case.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students brainstorm games and other applications of 2D arrays.
- Have students code a games using 2D arrays.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 09: Inheritance

Content Area: **Mathematics**
Course(s):
Time Period: **March**
Length: **8 blocks**
Status: **Awaiting Review**

Course Description & Instructional Notes

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through inheritance. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called subclasses. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a superclass and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Prior Knowledge:

Instructional Notes:

- **Building Computational Thinking Practices**
    - Students can design hierarchies by listing the attributes and behaviors for each object and pulling common elements into a superclass, leaving unique attributes and behaviors in the subclass. By creating a hierarchy system, students only need to write common program code one time, reducing potential errors and implementation time. This also allows for changes to be made more easily, because they can be made at the superclass level in the hierarchy and automatically apply to subclasses. During the development of a program, programmers often use comments to describe the behavior of a given segment of program code and to describe the initial conditions that are used. Students who develop the skill of explaining why program code does not work, such as methods being overloaded improperly or superclass objects attempting to call subclass methods, are much better equipped to foresee and avoid these hierarchy issues.

    - **Preparing for the AP Exam** One question on the free-response section of the exam will require students to write a class. This class could be part of an inheritance relationship. When overriding superclass methods in a subclass, method signatures must be the same. This includes the number, type, and order of any parameters of the overridden method. It is important for students to recognize when a method should be overridden, as well as when they can and should use methods from the superclass. Students who duplicate code unnecessarily will not earn full points on this question. Students will be asked to analyze program code that uses inheritance. In many cases, students struggle with determining whether a method is available to be called by an object of a class. When a method is called on a subclass object, the method that is executed is determined during run-time. If the subclass does not contain the called method, the superclass method will automatically be executed.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

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?

Student Learning Objectives

Students will be able to...

Create an inheritance relationship from a subclass to the superclass.

Define reference variables of a superclass to be assigned to an object of a subclass in the same hierarchy.

Call methods in an inheritance relationship.

Call Object class methods through inheritance.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

object - Objects do the action in an object-oriented program. An object can have things it knows (attributes) and things it can do (methods). An object is created by a class and keeps a reference to the class that created it.

class - A class defines what all objects of that class know (attributes) and can do (methods). You can also have data and behavior in the object that represents the class (class instance variables and methods). All objects of a class have access to class instance variables and class methods, but these can also be accessed using className.variable or className.method().

inheritance - One class can inherit object instance variables and methods from another. This makes it easy to reuse another class by extending it (inheriting from it). This is called specialization. You can also pull out common instance variables and/or methods from several related classes and put those in a common parent class. This is called generalization.

polymorphism - The runtime type of an object can be that type or any subclass of the declared type. All method calls are resolved starting with the class that created the object. If the method isn't found in the class that created the object, then it will look in the parent class and keep looking up the inheritance tree until it finds the method. The method must exist, or the code would not have complied.

parent class - One class can inherit from another and the class that it is inheriting from is called the parent class. The parent class is specified in the class declaration using the extends keyword followed by the parent class name.

child class - The class that is doing the inheriting is called the child class. It inherits access to the object instance variables and methods in the parent class.

subclass - A child class is also called a subclass.

superclass - A parent class is also called a superclass.

declared type - The type that was used in the declaration. List aList = new ArrayList() has a declared type of List. This is used at compile time to check that the object has the methods that are being used in the code.

run-time type - The type of the class that created the object. List aList = new ArrayList() has a run-time type of ArrayList. This is used at run-time to find the method to execute.

overrides - A child class can have the same method signature (method name and parameter list) as a parent class. Since methods are resolved starting with the class that created the object, that method will be called instead of the inherited parent method, so the child method overrides the parent method.

overload - At least two methods with the same name but different parameter lists. The parameter lists can differ by the number of parameters and/or the types.

getter - A method that returns the value of an instance variable in an object.

setter - A method that sets the value of am instance variable in an object.

accessor - Another name for a getter method - one that returns the value of a instance variable.

mutator - Another name for a setter method - one that changes the value of a instance variable.

extends - Used to specify the parent class to inherit from. It is followed by the name of the parent class, like this: public class ChildName extends ParentName. If no extends keyword is used in the class declaration, then the class will automatically inherit from the Object class.

static - Keyword used to indicate that a instance variable or method is part of the class and not part of each object created by the class.

super - Keyword used to call a method in a parent class. This is useful if a child class overrides an inherited method, but still wants to call it.

Planned Learning Experiences

Activating prior knowledge Have students review what they know about classes, methods, and the scope of variables by having them write a class based on specifications that can easily be extended by subclasses. This class will become the superclass for subclasses they write later in the unit.

Create a plan Given a class design problem that requires the use of multiple classes in an inheritance hierarchy, students identify the common attributes and behaviors among these classes and write these into a superclass. Any additional information that does not belong in the superclass will be categorized to determine the additional classes that might be necessary and what methods will need to be added or overridden in the subclasses.

Think aloud Provide students with a code segment that contains method calls using the super keyword. Have students describe the code segment out loud to themselves. Give students several individual statements that attempt to interact with the given code segment, and have them talk through each one, describing which statements would work and which ones would not, as well as the reasons why those statements wouldn't work.

Student response system Provide students with several statements where objects are created and the reference type and object type are different but related. Then provide students with calls to methods on these created objects. Use a student response system to have students determine whether each statement is legal, would result in a compile-time error, or would result in a run-time error.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher’s Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them;

and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.

MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students brainstorm real life examples of Inheritance and create a chart to represent the hierarchy of the classes.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 10: Recursion

Content Area: **Mathematics**
Course(s):
Time Period: **March**
Length: **3**
Status: **Awaiting Review**

Course Description & Instructional Notes

5-7.5% AP Exam Weighting

Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called recursion, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

Prior Knowledge:

Instructional Notes:

- **Building Computational Thinking Practices**
  - To better understand how recursion works, students should spend time writing their own recursive methods. Often, this can be overwhelming for students. One way to scaffold this skill for students is to require them to write a portion of program code, such as the base case, that can be used to complete the recursive method. Students should be able to determine the result of recursive method calls. Tracing the series of calls is a useful way to glean what the recursive method is doing and how it is accomplishing its purpose. Recursive algorithms, such as sorting and searching algorithms, often produce a result much more quickly than iterative solutions. Students also need to understand how many times statements in a recursive solution execute based on given input values.

- **Preparing for the AP Exam**
  - Recursion is primarily assessed through the multiple-choice section of the exam. Students are often asked to determine the result of a specific call to a recursive method or to describe the behavior of a recursive method. A call to a recursive method is just like a call to a nonrecursive method. Because a method is an abstraction of a process that has a specific result for each varied input, using a specific input value provides an understanding of how the method functions for that input. By understanding several instances of the method call, students can abstract and generalize the method's overall purpose or process. While students will not be required to write a recursive solution in the free-response section, recursive solutions are often a more straightforward way of writing the solutions than iterative designs. Writing recursive solutions and analyzing calls to recursive methods help engage students with all aspects of recursive methods and provide them with a deeper understanding of how recursion works.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

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.

Essential Questions

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?

Student Learning Objectives

Students will be able to...

Determine the result of executing recursive methods.

Apply recursive search algorithms to information in String, ID array, or ArrayList objects.

Apply recursive algorithms to sort elements of array or ArrayList objects.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

base case - A way to stop the recursive calls. This is a return without a recursive call.

call stack - A class defines what all objects of that class know (fields) and can do (methods). You can also have data and behavior in the object that represents the class (class fields and methods). All objects of a class have access to class fields and class methods, but these can also be accessed using className.field or className.method().

recursive method - A method that contains at least one call to itself inside the method.

Planned Learning Experiences

Sharing and responding Provide students with the pseudocode to multiple recursive algorithms, and have students write the base case of the recursive methods and share it with their partner. The partner should then provide feedback, including any corrections or additions that may be needed.

Look for a pattern Provide students with a recursive method and several different inputs. Have students run the recursive method, record the various outputs, and look for a pattern between the input and related output. Ask students to write one or two sentences as a broad description of what the recursive method is doing.

Code tracing When looking at a recursive method to determine how many times it executes, have students create a call tree or a stack trace to show the method being called and the values of any parameters of each call. Students can then count up the number of times a statement executes or a method is called.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students create execution counts for multiple levels of recursive processes and find a pattern.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference

Modification: English Language Learners

- Pair programming with another student
- Print out slides for students to reference

Unit 11: Review for the AP Exam

Content Area: **Mathematics**
Course(s):
Time Period: **Marking Period 3**
Length: **20 blocks**
Status: **Published**

Instructional Notes

In this unit, students wlll students will get a practice exam in the same format as the AP Computer Science in Java test and complete both multiple choice and free response questions from previously administered exams. Students will also be given mini-lessons to review concepts as needed, as well as online video review instruction.

Prior Knowledge:

Students should be able to recall any topic from the course with minimal reminders.

Instructional Notes:

At this point, the students have learned all of the JAVA they need to know for the AP Exam. Formative assessing should guide differentiated insructional decisions with regard to reviewing particular topics. CodeHS has a review course available to prepare for the AP exam- this can be a good guiding structure for review. Using materials from previous AP Exams, especially the student samples, scoring guidelines, and chief reader reports available at collegeboard.com would greatly benefit students as they prepare.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

We can prepare for novel problems by experiencing new ones.

Essential Questions

How can we prepare for an exam?

Student Learning Objectives

Students will able to:

Explain the format of the AP Computer Science A exam

Identify course topics that they have a strong understanding of

Identify course topics that they need to review

Vocabulary & Learning Experiences

Essential Academic Vocabulary

none

Planned Learning Experiences

The AP Computer Science A framework details the concepts and skills students must master to be successful on the AP exam. To address those concepts and skills effectively, it helps to incorporate a variety of instructional approaches and best practices into daily lessons and activities. The following list presents strategies that can help students develop mastery of the skills by engaging them in learning activities that apply their understanding of course concepts.

Programming and Problem-Solving Strategies:

- Code Tracing
- Create a Plan
- Error Analysis
- Identify a subtask
- Look for a Pattern
- Marking the Text
- Pair Programming
- Predict and compare
- Simplify the problem
- 'Think Aloud

Cooperative Learning Strategies:

- Ask the expert
- Discussion Group
- Jigsaw
- Kinisthetic learning
- Sharing an responding

- Student Response system
- Think-pair-share
- Unplugged activities
- Using manipulatives

Making Connections Strategies:

- Activating prior knowledge
- Diagramming
- Bote-taking
- Paraphrasing
- Quickwrite
- Vocabulary Organizer

These strategies above should be implemented throughout the course as appropriate to the specific skill being learned. As a guideline, college board in the AP Computer Science A Course and Exam Description have created a suggested list of activities and strategies to tie along with the skills of the course. In using this guideline [pgs 170-179], the students will develop compuational thinking practictes that are fundamental to the discipline of computer science as well as successful completion of the AP Exam.

Resources

- Code HS
- Google Classroom
- AP Classroom
- JavaNotes
- atomic toddler
- Practice It!
- Codingbat

AP Review Specific Resources

AP Central: Important Advice, Teacher's Guide, Exam Info and FRQs, and Labs

Computer Science Resource Bank,

Java Review for the AP CS A Exam

This great resource, written by Barbara Ericson, is loaded with review questions and clear explanations.

Java Review

Assessments

Formative Assessments

Java Post Test

CodeHS Review Course

CodeHS 25 question unit quizzes

Runestone Academy Review course

Warm Ups and Exit Tickets as needed (based on student feedback)

Summative Assessments

AP Exam

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, “Does this make sense?” They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7×8 equals the well remembered $7 \times 5 + 7 \times 3$, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2×7 and the 9 as $2 + 7$. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Special Education

- Refer to the College Boards accommodations for SPED students, and provide students with the appropriate accommodations for taking the AP exam.
- If students are not actively preparing for the actual exam, give students as much time as they'd like to complete the exercises.

Modification: English Language Learners

- Refer to the College Boards accommodations for ELL students, and provide students with the appropriate accommodations for taking the AP exam.
- If students are not actively preparing for the actual exam, give students as much time as they'd like to complete the exercises.

Unit 12: Projects using our Coding Skills

Content Area: **Mathematics**
Course(s):
Time Period: **Marking Period 4**
Length: **10 blocks**
Status: **Published**

Instructional Notes

In this unit, students wlll have a chance to build your their own application using the new skills they've learned!

Prior Knowledge:

Students should know about:

- Basic Java
- Methods
- Classes and OOP
- Data Structures
- Algorithms and Recursion

Instructional Notes:

Students should be encouraged to solve a problem they experience in real life and write an app to solve that problem. Students should be given extensive freedom in grouping and choice of problem to address, but teacher should plan for intermediary due dates to make sure students are progressing in a timely manner. This unit is a chnace for students to experience what a career in programming could be like.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

Enduring Understandings

Our programming skills can be used to solve real-life problems.

Essential Questions

Why did we learn JAVA?

Student Learning Objectives

Student will be able to:

Synthesize concepts and skills learned in the course to create their own final project.

Scope their project (eliminate features that aren't necessary) so that it fits in the timeframe allotted.

Present their project to their classmates and talk about how the project was developed.

Vocabulary & Learning Experiences

Essential Academic Vocabulary

none

Planned Learning Experiences

The AP Computer Science A framework details the concepts and skills students must master to be successful on the AP exam. To address those concepts and skills effectively, it helps to incorporate a variety of instructional approaches and best practices into daily lessons and activities. The following list presents strategies that can help students develop mastery of the skills by engaging them in learning activities that apply their understanding of course concepts.

Programming and Problem-Solving Strategies:

- Code Tracing
- Create a Plan
- Error Analysis
- Identify a subtask
- Look for a Pattern
- Marking the Text
- Pair Programming
- Predict and compare
- Simplify the problem
- 'Think Aloud

Cooperative Learning Strategies:

- Ask the expert
- Discussion Group
- Jigsaw
- Kinisthetic learning
- Sharing an responding

- Student Response system
- Think-pair-share
- Unplugged activities
- Using manipulatives

Making Connections Strategies:

- Activating prior knowledge
- Diagramming
- Bote-taking
- Paraphrasing
- Quickwrite
- Vocabulary Organizer

These strategies above should be implemented throughout the course as appropriate to the specific skill being learned. As a guideline, college board in the AP Computer Science A Course and Exam Description have created a suggested list of activities and strategies to tie along with the skills of the course. In using this guideline [pgs 170-179], the students will develop compuational thinking practictes that are fundamental to the discipline of computer science as well as successful completion of the AP Exam.

Resources

- Code HS
- AP Classroom
- JavaNotes
- atomic toddler
- Practice It!
- Codingbat

Assessments

Formative Assessments

Informal student progress checks

Summative Assessments

Final Project and Presentation

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7 × 8 equals the well remembered 7 × 5 + 7 × 3, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2 × 7 and the 9 as 2 + 7. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 – 3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modification: Special Education

- Have students work in pairs.
- Utilize *Gantt* Chart handout, and scaffold components of the final project.
- Pay careful attention to the scope of the project, so students are able to complete by the end of the school year.

Modification: English Language Learners

- Have students work in pairs.
- Utilize *Gantt* Chart handout, and scaffold components of the final project.
- Pay careful attention to the scope of the project, so students are able to complete by the end of the school year.

assessed on the AP Exam, this content exists in the program code of nearly every assessment question. It is important for students to spend adequate time practicing writing expressions and applying meaning to specific operators during this unit. Early success will build students' confidence, which will be necessary as we build on this knowledge, adding new concepts and requiring more sophisticated application of the concepts.

Technology Integration:

Computer Science naturally integrates technology on a daily basis.

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 more quickly and with a greater degreee 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 sequences 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 the mathematical concepts being used in the programs and apps that you use most often?

Student Learning Objectives

Students will be able to:

- call system class methods to generate output to the console
- recognize the difference between display behavior of `System.out.print` and `System.out.println`
- create string literals

- declare, initialize and assign a value to a variable
- identify the most appropriate data type category for a particular specification

- categorize a data type as either primitive or reference
- declare a final variable

- evaluate arithmetic expressions in a program code
- use the modulus operator (`%`)

- understanding of basic operations (`+`,`-`,`/`, `*`, `%`) in program code
- how to declare, assign and initialize variables in Java

- import and initialize a new Scanner to take in user input
- create variables that take the assigned value of the user input.
- choose the correct command that will allow the program to receive the input value that corresponds with its desired data type

- Evaluate arithmetic expressions that use casting
- Understand that integer values in Java are stored using a finite amount (4 bytes) of memory. Therefore, an `int` value must be in the range from `Integer.MIN_VALUE` to `Integer.MAX_ VALUE` inclusive
- Understand when an expression evaluates to an int value outside of the allowed range, an integer overflow occurs
- Recognize and use implicit casting
- Use casting to round to the nearest integer

Vocabulary & Learning Experiences

Essential Academic Vocabulary

Compiler - Software that translates the Java source code (ends in .java) into the Java class file (ends in .class).

Compile time error - An error that is found during the compilation. These are also called syntax errors.

Main Method - Where execution starts in a Java program.

Boolean - An expression that is either true or false.

Camel Case - One way to create a variable name by appending several words together and uppercasing the first letter of each word after the first word (myScore).

Casting a Variable - Changing the type of a variable using (type) name.

Double - A type in Java that is used to represent decimal values like -2.5 and 323.203.

Declare a Variable - Specifying the type and name for a variable. This sets aside memory for a variable of that type and associates the name with that memory location.

Initializing a Variable - The first time you set the value of a variable.

Integer - A whole number like -32 or 6323.

modulo - The % operator which returns the remainder from one number divide by another.

Operator - Common mathematical symbols such as + for addition and * for multiplication.

Shortcut Operators - Operators like x++ which means x = x + 1 or x *=y which means x = x * y.

Variable - A name associated with a memory location in the computer.

boolean - used to declare a variable that can only have the value true or false.

double - used to declare a variable of type double (a decimal number like 3.25).

false - one possible value for a boolean variable.

int - used to declare a variable of type integer (a whole number like -3 or 235).

static - means that the field or method exists in the object that defines the class.

true - one possible value for a boolean variable.

Planned Learning Experiences

- Error analysis Provide students with code that contains syntax errors. Ask students to identify and correct the errors in the provided code. Once they feel they have identified and corrected all syntax errors, have them verify their conclusion by using a compiler and an IDE that does not autocorrect errors.

- Activating prior knowledge The basic arithmetic operators of +, −, /, and * are similar to what students have experienced in math class or when using a calculator. Give students a list of expressions and ask them to apply what they know from math class to evaluate the meaning of the expressions. Have them verify their results by putting them into a compiler.

- Sharing and responding Put student into groups of two. Provide each student with a different set of statements; in each pair, one student should have a list of statements that contain compound assignment operators, while the other student should have a list of statements that accomplish the same thing without using compound statements. Be sure the statements are in a different order. Students should take turns describing what a statement does to their partner, and the partner should determine which statement of theirs is equivalent to the one being described.

- Predict and compare Provide students with several statements that involve casting. Each cast should be on a different value in the statement. Have students predict the resulting value. For any statements that would not compile or work as intended, have students explain the problem and propose a solution. They should verify their results by putting those results into a compiler.

Resources

Teaching and Learning Resources

- CodeHS
- AP Classroom
- codestepbystep
- Practice It!
- Codingbat
- Runestone Academy: csawesome

AP Review Specific Resources

AP Central: Important Advice, Teacher’s Guide, Exam Info and FRQs, and Labs

Assessments

Formative Assessments

Quizzes embedded in CodeHS Modules and Code Review

Summative Assessments

Unit Quizzes (multiple choice only)

Student Choice Unit Project

NJSLS Standards - Mathematics

NJSLS Standards in Mathematics Copied and Pasted as well as linked.

NJSLS Standards - Mathematics

Standards for mathematical practices

1 Make sense of problems and persevere in solving them.

Mathematically proficient students start by explaining to themselves the meaning of a problem and looking for entry points to its solution. They analyze givens, constraints, relationships, and goals. They make conjectures about the form and meaning of the solution and plan a solution pathway rather than simply jumping into a solution attempt. They consider analogous problems, and try special cases and simpler forms of the original problem in order to gain insight into its solution. They monitor and evaluate their progress and change course if necessary. Older students might, depending on the context of the problem, transform algebraic expressions or change the viewing window on their graphing calculator to get the information they need. Mathematically proficient students can explain correspondences between equations, verbal descriptions, tables, and graphs or draw diagrams of important features and relationships, graph data, and search for regularity or trends. Younger students might rely on using concrete objects or pictures to help conceptualize and solve a problem. Mathematically proficient students check their answers to problems using a different method, and they continually ask themselves, "Does this make sense?" They can understand the approaches of others to solving complex problems and identify correspondences between different approaches.

2 Reason abstractly and quantitatively.

Mathematically proficient students make sense of quantities and their relationships in problem situations. They bring two complementary abilities to bear on problems involving quantitative relationships: the ability to decontextualize—to abstract a given situation and represent it symbolically and manipulate the representing symbols as if they have a life of their own, without necessarily attending to their referents—and the ability to contextualize, to pause as needed during the manipulation process in order to probe into the referents for the symbols involved. Quantitative reasoning entails habits of creating a coherent representation of the problem at hand; considering the units involved; attending to the meaning of quantities, not just how to compute them; and knowing and flexibly using different properties of operations and objects.

3 Construct viable arguments and critique the reasoning of others.

Mathematically proficient students understand and use stated assumptions, definitions, and previously established results in constructing arguments. They make conjectures and build a logical progression of statements to explore the truth of their conjectures. They are able to analyze situations by breaking them into cases, and can recognize and use counterexamples. They justify their conclusions, communicate them to others, and respond to the arguments of others. They reason inductively about data, making plausible arguments that take into account

the context from which the data arose. Mathematically proficient students are also able to compare the effectiveness of two plausible arguments, distinguish correct logic or reasoning from that which is flawed, and—if there is a flaw in an argument—explain what it is. Elementary students can construct arguments using concrete referents such as objects, drawings, diagrams, and actions. Such arguments can make sense and be correct, even though they are not generalized or made formal until later grades. Later, students learn to determine domains to which an argument applies. Students at all grades can listen or read the arguments of others, decide whether they make sense, and ask useful questions to clarify or improve the arguments.

4 Model with mathematics.

Mathematically proficient students can apply the mathematics they know to solve problems arising in everyday life, society, and the workplace. In early grades, this might be as simple as writing an addition equation to describe a situation. In middle grades, a student might apply proportional reasoning to plan a school event or analyze a problem in the community. By high school, a student might use geometry to solve a design problem or use a function to describe how one quantity of interest depends on another. Mathematically proficient students who can apply what they know are comfortable making assumptions and approximations to simplify a complicated situation, realizing that these may need revision later. They are able to identify important quantities in a practical situation and map their relationships using such tools as diagrams, two-way tables, graphs, flowcharts and formulas. They can analyze those relationships mathematically to draw conclusions. They routinely interpret their mathematical results in the context of the situation and reflect on whether the results make sense, possibly improving the model if it has not served its purpose.

5 Use appropriate tools strategically.

Mathematically proficient students consider the available tools when solving a mathematical problem. These tools might include pencil and paper, concrete models, a ruler, a protractor, a calculator, a spreadsheet, a computer algebra system, a statistical package, or dynamic geometry software. Proficient students are sufficiently familiar with tools appropriate for their grade or course to make sound decisions about when each of these tools might be helpful, recognizing both the insight to be gained and their limitations. For example, mathematically proficient high school students analyze graphs of functions and solutions generated using a graphing calculator. They detect possible errors by strategically using estimation and other mathematical knowledge. When making mathematical models, they know that technology can enable them to visualize the results of varying assumptions, explore consequences, and compare predictions with data. Mathematically proficient students at various grade levels are able to identify relevant external mathematical resources, such as digital content located on a website, and use them to pose or solve problems. They are able to use technological tools to explore and deepen their understanding of concepts.

6 Attend to precision.

Mathematically proficient students try to communicate precisely to others. They try to use clear definitions in discussion with others and in their own reasoning. They state the meaning of the symbols they choose, including using the equal sign consistently and appropriately. They are careful about specifying units of measure, and labeling axes to clarify the correspondence with quantities in a problem. They calculate accurately and efficiently, express numerical answers with a degree of precision appropriate for the problem context. In the elementary grades, students give carefully formulated explanations to each other. By the time they reach high school they have learned to examine claims and make explicit use of definitions.

7 Look for and make use of structure.

Mathematically proficient students look closely to discern a pattern or structure. Young students, for example, might notice that three and seven more is the same amount as seven and three more, or they may sort a collection of shapes according to how many sides the shapes have. Later, students will see 7×8 equals the well remembered $7 \times 5 + 7 \times 3$, in preparation for learning about the distributive property. In the expression x2 + 9x + 14, older students can see the 14 as 2×7 and the 9 as $2 + 7$. They recognize the significance of an existing line in a geometric figure and can use the strategy of drawing an auxiliary line for solving problems. They also can step back for an overview and shift perspective. They can see complicated things, such as some algebraic expressions, as single objects or as being composed of several objects. For example, they can see 5 –

3(x – y) 2 as 5 minus a positive number times a square and use that to realize that its value cannot be more than 5 for any real numbers x and y.

8 Look for and express regularity in repeated reasoning.

Mathematically proficient students notice if calculations are repeated, and look both for general methods and for shortcuts. Upper elementary students might notice when dividing 25 by 11 that they are repeating the same calculations over and over again, and conclude they have a repeating decimal. By paying attention to the calculation of slope as they repeatedly check whether points are on the line through (1, 2) with slope 3, middle school students might abstract the equation (y – 2)/(x – 1) = 3. Noticing the regularity in the way terms cancel when expanding (x – 1)(x + 1), (x – 1)(x2 + x + 1), and (x – 1)(x3 + x2 + x + 1) might lead them to the general formula for the sum of a geometric series. As they work to solve a problem, mathematically proficient students maintain oversight of the process, while attending to the details. They continually evaluate the reasonableness of their intermediate results.

NJSLS 2020- Computer Science and Design Thinking

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.4	Model with mathematics.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
MA.K-12.1	Make sense of problems and persevere in solving them.

Modifications/Accommodations

Modification: Advanced

- Have students explore escape sequences such as `\n` and `\t`. What effect to they have in string literals?
- Have students explore exponents in Java (`Math.pow(a, b);` = a^b)
- Have students research how many bytes a `String` takes up.

Modification: Special Education

- Pair programming with another student
- Print out slides for students to reference
- Allow use of calculator

Modification: English Language Learners

- Pair programming with another student

- Print out slides for students to reference