Mathematics Department

Introduction to Computer Programming

Developed by: Monika Truty

Effective Date: September 2023

Scope and Sequence

Month	
September	**Python: Basic Python and Console Interaction** -Variables -User Input -Mathematical and String Operators **Python: Conditionals** -Booleans -If Statements -Comparison and Logical Operators -Floating Point Numbers and Rounding
October	**Python: Looping** -While and For Loops (nested) -Break and Continue **Python: Functions and Exceptions** -Functions, Parameters, Return Values -Exceptions
November	**Python: Strings** -Indexing and Slicing -Immutability -String Methods **Python: Creating and Altering Data Structures** -Tuples, Lists -Loops and Lists (traversing) -List Methods
December	**CSEducation Week Activities** **Python: Extending Data Structures**

	-2D lists -List Comprehension -Packing and Unpacking -Dictionaries -Equivalence and Identity
January	**Python: Guess The Word Project** -Apply the learned project
February	**Python: Introduction to Programming with Turtle Graphics** -Turtle commands to move Turtle -For/While Loops, If/Else Statements -Creating Graphics with all structures learned **ASSESSMENT (Python Final)**
March	**Java: Basic Java** -Variables -User Input -Booleans and Conditionals
April	**Java: Basic Java (cont.)** -Logical and Comparison Operators -For and While Loops -DeMorgan’s Laws
May	**Java: Methods** -Java Methods, Parameters, Return Values -String Methods -Exceptions -String Processing
June	**Java: Classes and Object-Oriented Programming** -Classes -Objects -Getter and Setter Methods

Unit 1

Python: Basic Python and Console Interaction

Summary and Rationale

Students will learn the basics of programming by simple sequential writing programs that interact with users through the keyboard. Basic understanding and structure of variables, variable types must be applied. This lays the foundation to the course, and students begin to develop logical thinking practices. Students will be building interactive programs utilizing string or integer user input prompts. Students will learn the difference between mathematical and string operators, and how to document their programs.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)	
AP - Algorithms & Programming	
2-AP-17	Systematically test and refine programs using a range of test cases.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
CS - Computing Systems	
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
Computer Science & Design Thinking Practices	

Fostering an Inclusive Computing and Design Culture	
Collaborating Around Computing and Design	
Interaction of Technology and Humans	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.

W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus

Enduring Understandings:	**Essential Questions:**
A program is a sequence of commands that are executed or run by a computer. Programmers can print messages (strings), numbers (integers), or variables, and their types. Variables are placeholders for values. Variables can store strings, integers, booleans, or floats. User Interface is the inputs and outputs that allow a user to interact with a piece of software. User input can include a variety of forms such as clicks, keyboard keys, or mouse overs. Inputs are data that are sent to a computer for processing by a program. Outputs are any data that are sent from a program to a device. Outputs can come in a variety of forms, such as audio, visuals, or text. Mathematical operators are used in programming to do operations on numerical values or to join string values. Operators in Python have a specific operator precedence that programmers need to consider. Comments are used in programs as notes or documentation and do not affect how the program executes. Metadata is the top part of our programs, and includes main comments about the program and all defined variables.	What is a computer program? Why is it useful for programs to be able to accept user input? Describe what the modulus operator (%) does. What would be the result of 8 % 3? Why is it useful for programs to be able to perform mathematical computations? How can you use the addition operator, "+", with strings?

Evidence of Learning (Assessments)
Tests Quizzes Homework Programs Projects

Class participation

Objectives (SLO)

Students will know:

- Printing in Python
- Variables and Types
- User Input
- Mathematical and String Operators
- Documentation/Comments

Students will be able to:

- Print text in Python
- Define Python variables and basic types (string, integer, boolean, float)
- Incorporate user input into their programs
- Convert between variable types
- Create programs that use basic math to compute useful things
- Create programs that take in user input, do simple computations with the input, and produce useful output
- Use mathematical operators with strings
- Incorporate comments into their programs in order to make them more readable

.

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions

- Frequently check on progress of independent work

504:

Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 2

Python: Conditionals & Looping

Summary and Rationale

Students will learn about how programs make decisions and how to build efficient programs. Efficient programs are those that have a quick runtime, and avoid repeated code. If and if/else statements, for and while loops are all structures that make programs execute efficiently. Comparison operators give the ability to compare two values. Using comparison operators allows programs to make decisions. Logical operators give the ability to connect or modify Boolean expressions. Students will learn about break and continue commands that allow the program to break out of a loop at a certain time.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

AP - Algorithms & Programming

2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.

Computer Science & Design Thinking Practices

Fostering an Inclusive Computing and Design Culture

Collaborating Around Computing and Design	
Developing and Using Abstractions	
Interaction of Technology and Humans	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	

RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus	
Enduring Understandings:	**Essential Questions:**
Booleans refer to a value that is either true or false. Named after English-born mathematician, philosopher and logician, George Boole, Booleans are used to test whether a condition is true or false. If and if/else statements allow programs to make a decision. If some condition is true, then a series of instructions will execute. However if the initial condition is false, if there exists an else if statement, the program continues checking until a condition is true. If no conditions are true and an else statement was not used, nothing will execute. If all conditions are false and an else statement exists at the end of the structure, the else statement runs. This changes the flow of our programs. If/Else statements allow programmers to use conditions to determine how their code should run. Comparison operators (<, <=, >, >=, ==, !=) are used to compare two values. These values could be variables, and can have a string, integer, float, or boolean type. Logical operators or (||), and (&&), and NOT (!) are used to help represent a situation or condition in an if statement. Logical operators are used to compare conditions. Primitive data types that are of numerical values can be rounded through specific commands. If those commands are being used then the programmer can determine the rounded decimal spot. When using comparison operators with floating point numbers in Python, we will sometimes see strange behavior because of bizarre rounding methods. When using a comparison operator with floating point numbers, we should use round(x) to avoid strange rounding behavior. round(x, n) will round the float x to n decimal places. round(x) will round the float x to 0 decimal places.	When would a programmer decide to use an if statement and when would a programmer use an if/else statement? Explain the difference between = and ==. What is the main difference between a for and while loop? When would a programmer use a break statement? If a programmer is running an infinite loop, where would a programmer begin to debug the loop?

While loops allow code to be executed repeatedly based on a condition. They are also reminded of the possibility of creating an infinite loop, which occurs if the exit condition of the while loop is never met, causing the code inside the while loop to repeat continuously. For loops repeat instructions for a fixed number of times, controlled by the value i.

A break statement is used to immediately terminate a loop. A continue statement is used to skip out of future commands inside a loop and return back to the top of the loop. These statements can be used with for or while loops.

Control structures within control structures, which are referred to as nested control structures.

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know:

- Booleans
- If Statements
- Comparison & Logical Operators
- Floating Point Numbers and Rounding
- For & While Loops, and nested control structures
- Break and Continue

Students will be able to:

- use if/else statements for control flow in programs
- understand the comparison operators (<, <=, >, >=, ==, !=) and apply them in programs to compare values
- describe the meaning and usage of each logical operator: or (||), and (&&), and NOT (!) and use them as boolean values in conditions and variables
- to use floating point numbers and round values in their programs
- Effectively use for and while loops in their programs
- Detect and resolve infinite loops
- Describe why a break or continue statement would be needed in a coding scenario
- Combine control structures to solve complicated problems

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 3

Python: Functions and Exceptions

Summary and Rationale

Students will learn about functions, how to use parameters and return values. As functions are being used in programs, students will explore where variables exist and what the difference is between a local and global variable. Students will also learn about exceptions and how exceptions are used to override certain error codes in Python.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

AP - Algorithms & Programming

3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

Computer Science & Design Thinking Practices

Developing and Using Abstractions

Interaction of Technology and Humans

Career Readiness, Life Literacies and Key Skills

9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.

9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus

Enduring Understandings:	**Essential Questions:**

A function is a named group of programming instructions. Functions let us break our program into different parts that we can organize and reuse however we like. Functions are the main building block of complex Python programs.

Functions should only be defined once in a program.
A function can be used in a program with a function Call, which is a command that executes the code within a function. Functions can be called many times in a program.

Variables used in a function can have a global or local scope. This is determined by where the variable was defined and where it is being reassigned. Global variables are permanent and can be used anywhere in a program. Local variables are temporary and can be used only in the part of the code where it was created, like inside an event handler. The local variables are “deleted” once the event handler or function is done running.

Exceptions are used in programming when the user types in an input but the program can’t process this data. For example, if the program is a calculator and the user tries to divide any integer by another integer. If the divisor is 0, the program will stop and show a ZeroDivisionError. This can be overridden so the program runs, and only prints a message stating that the user can’t divide by zero, to try again. The exception would be programmed for that specific error code, so that if this occurs, the program doesn’t just stop, it continues running.

Why is using a function in a program beneficial for the programmer and how does it contribute to the efficiency of the program?

How can functions make programs easier to write?

When do programs require an exception?

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know:
- Functions
- Parameters
- Namespaces in Functions
- Return Values
- Exceptions

Students will be able to:
- Effectively use parameters to customize functions in their programs
- describe the different namespaces with regards to variables and functions
- remove complexity from their programs by abstracting with functions

- create programs that can gracefully handle exceptions

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS
Please select all standards that apply to this unit of study: Act as a responsible and contributing community members and employee Attend to financial well-being Consider the environmental, social and economic impacts of decisions Demonstrate creativity and innovation Utilize critical thinking to make sense of problems and persevere in solving them Model integrity, ethical leadership and effective management Plan education and career paths aligned to personal goals Use technology to enhance productivity increase collaboration and communicate effectively Work productively in teams while using cultural/global competence Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 4

Python: Strings

Summary and Rationale

Students will learn how to work with strings in Python. If given a string value like “computer”, students will learn how to index each character, slice the string by the index, and lastly learn to execute string methods on the string. This unit is structuring the knowledge needed for the final Python project.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

AP - Algorithms & Programming

3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3B-AP-11	Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

Computer Science & Design Thinking Practices

Developing and Using Abstractions

Interaction of Technology and Humans

Career Readiness, Life Literacies and Key Skills

9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.
Instructional Focus	

Enduring Understandings:

An index is every letter, space, symbol in a string is assigned this number and used as a reference in programming. Commonly said, this number is used to access a letter or set of letters in programming. The first letter on the left is always assigned the index value 0, followed by the second or index 1, third or index 2, fourth or index 3, etc.

Data types can be mutable or immutable. Not being able to change something without replacing it entirely, like strings, are immutable. Mutable means changeable. This means that the values stored at each index can be changed at that specific value.

Using the length of the string can be helpful, because a for loop can be used to traverse the string. Using the in keyword, a programmer can use if statements to check to see if an element is IN the string or for loops to check to see for every element IN the string, some kind of commands will execute.

Essential Questions:

Why is it helpful using string indexing?

Why are strings immutable?

In what ways can we use the in keyword with strings and data structures?

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know:

- Indexing
- Slicing
- Immutability
- Strings and For Loops
- String Methods

Students will be able to:

- use indexing in order to find a specific character in a string
- use slicing to select a set of values from a string
- explain what immutability is and how this applies to strings in Python
- iterate over characters in a string using for loops
- use the in keyword to check if a character is in a string
- use various string methods to alter string values

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student's 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals

Use technology to enhance productivity increase collaboration and communicate effectively Work productively in teams while using cultural/global competence Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 5

Python: Creating, Altering, Extending Data Structures

Summary and Rationale

Students will learn about and practice using tuples, lists, 2d lists, and dictionaries. All of these structures are more ways on how to store more than one value or type of data. There is a key difference between tuples, lists, dictionaries, and strings. Students explore these in this unit.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

DA - Data & Analysis

3A-DA-12	Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.

AP - Algorithms & Programming

3A-AP-14	Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-10	Use and adapt classic algorithms to solve computational problems.
3B-AP-23	Evaluate key qualities of a program through a process such as a code review.
Computer Science & Design Thinking Practices	
Recognizing and Defining Computational Problems	
Developing and Using Abstractions	
Fostering an Inclusive Computing and Design Culture	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.

Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus

Enduring Understandings:	**Essential Questions:**
A tuple is a heterogeneous, immutable data type that stores an ordered sequence of things that can be accessed using indices. A list is a mutable, heterogeneous data type that stores an ordered sequence of things. Methods, in general, are like functions that can be called on objects. When working with lists, there are different methods that can be applied so that there's an easier way to access certain elements. Indexing and slicing can be used to extract specific items from a list of lists. This allows us to store a list of lists, which allows us to store more information in one list. This can almost feel like a dataset of lists. Packing and unpacking makes assigning variables to list items very easy and quick. This means we can store multiple variables with the different elements in a certain list, in one simple line of code. Sometimes programmers are given values that have a specific key to each value. These are called dictionaries, and they store information as a set of keys and corresponding values. Dictionaries are helpful when storing data that is very independent. For example, a student's score can be the value, and the key to each score will be the student's last name. This makes it easier to store and find.	Which data structures are mutable and which are immutable? When is it applicable to use a dictionary instead of a list? What is the difference between equivalence and identity?

Two elements that LOOK the same but have different values are equivalent elements. Two elements that have the same value but may look different are identities.

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know:

- Tuples & Lists
- For Loops and Lists
- List Methods
- 2d Lists
- Packing and Unpacking
- Dictionaries
- Equivalence vs. Identity

Students will be able to:

- create and store information in tuples and lists
- use for loops to go through items in a list
- apply useful list methods to alter and access information about a list
- use 2d lists to store information in rows and columns
- pack and unpack lists in order to quickly and efficiently assign variables to list items
- use dictionaries to structure data
- predict if two values are identical and/or equivalent

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:

Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 6

Python: Introduction to Programming with Turtle Graphics

Summary and Rationale

Students will learn basic turtle programming while applying their knowledge of programming in Python. Students will begin with basic commands to move the turtle and will learn how to add color, iteration, and structure to their programs.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

AP - Algorithms & Programming

2-AP-17	Systematically test and refine programs using a range of test cases.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-19	Systematically design and develop programs for broad audiences by incorporating feedback from users.
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
3A-AP-23	Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

CS - Computing Systems	
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
Computer Science & Design Thinking Practices	
Recognizing and Defining Computational Problems	
Creating Computational Artifacts	
Fostering an Inclusive Computing and Design Culture	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.

Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus

Enduring Understandings:	**Essential Questions:**
The layout of Tracy's grid world is a coordinate plane where coordinate pairs are needed to locate the turtle. Turtle commands are needed to create Python graphics. Turtle commands include: forward(#), backward(#), left(degrees), right(degrees), clear(), pensize(#), circle(radius), circle(radius, deg, sides), setposition(x,y), penup(), pendown(), color("text"), speed(#), end_fill(), begin_fill().	How can the turtle draw diagonal lines? Why does indentation matter in Python? How can the turtle draw a perfect hexagon?
Evidence of Learning (Assessments)	
Tests Quizzes Homework Programs Projects Class participation	
Objectives (SLO)	
Students will know: ● Turtle Commands ● Turning Turtle using angles ● Turtle commands using variables, user input, functions, parameters, for & while loops, if/else statements	Students will be able to: ● use turtle commands to move, turn, reposition, and locate the turtle around the grid ● add some flair to their turtle graphics programs by controlling color, pensize, and fill ● use the variable i to control much more of their code by setting specific values

	● use top down design and to write programs that will solve complex problems ● use for/while loops and functions to help avoid repeated code

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 7

Java: Basic Java

Summary and Rationale

Students will learn the basic programming structures in Java. This includes calling and defining classes and using variables, user input, arithmetic operations, if statements, while and for loops, documentation, and logical and comparison operators. Java is object oriented, therefore there are different logical thinking skills that need to be acquired in order to understand the ways of Java. DeMorgan's laws will help in understanding how objects behave in future units. Strings are also handled differently in Java than Python. Primitive and nonprimitive types need to be defined and used.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)	
AP - Algorithms & Programming	
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance and explain the benefits and drawbacks of choices made.
3B-AP-21	Develop and use a series of test cases to verify that a program performs according to its design specifications.
Computer Science & Design Thinking Practices	
Recognizing and Defining Computational Problems	

Creating Computational Artifacts	
Fostering an Inclusive Computing and Design Culture	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus	
Enduring Understandings:	**Essential Questions:**
Variables allow us to store information such as numbers, words, or true/false expressions. A variable can be thought of as a box that stores information inside. In Java, variables are composed of three things: a name, type, and value. User input personalizes a program to the user. To retrieve user input, the programmer must plan what type of value they are requesting from the user. The various methods include readLine, readInt, readDouble, and readBoolean. This will allow the program to take in data from the user, making the experience more personable. Arithmetic Expressions allow us to perform mathematical operations within Java. Such expressions can be used for basic math and even more complex algorithms. Casting helps with converting values, or turning something of one type into another type. A boolean refers to a value that is true or false. Those are the only values of a boolean expression, and these are useful if we want to check if something is true or false. Logical operators allow us to connect or modify Boolean expressions. Three logical operators are the !, ||, && characters. ! = NOT || = OR && = AND Logical operators can be used in combination. With these logical operators, we can construct logical statements such as “I go to sleep when I am tired OR it’s after 9pm”, “I wear flip flops when I am outside AND it is NOT raining” Comparison operators let us compare two values. Using comparison operators in programming is similar to math in that less than <, greater than >, less than or equal to <=, and greater than or equal to >= are the same. The differences are that operators for equal to are == and not equal are !=. Using comparison operators allows our program to make decisions. For loops are used to repeat code a fixed number of times. While loops are a way to repeat a block of code so long as some condition remains true. The condition is written in the form of a boolean	How do we write code that tells us whether a user is logged in to our program? How do DeMorgan’s laws help us understand the logic of our program? What are primitive types? What are non primitive types?

expression. As long as the boolean expression remains true, code within the while loop will be executed. The moment that the boolean expression becomes false, code outside of the while loop will be executed; the loop is done.

Infinite loops can occur unintentionally if you are not careful with the conditions of a while loop. In these cases, the infinite loop can cause the program to crash. However, infinite loops can be a very useful tool in programming. If your program needs to repeat a block of code an indefinite number of times, an infinite loop may be the correct approach. Repeating code is helpful, but it’s just as important to be able to stop the loop so that the rest of the program can continue executing. Loops can be stopped using the break statement. When the loop encounters a break statement, it quits running the loop and program flow continues.

In a Short-Circuit Evaluation, if the result of a Boolean Expression can be determined by the first argument, the second argument is not evaluated.

De Morgan’s Laws are rules that show how we can negate “and”s and “or”s. Not (A and B) is the same as (not A) or (not B). Similarly, not (A or B) is the same as (not A) and (not B).

A String is a sequence of characters. We use Strings to represent full words and sentences. For example, the famous “Hello World” is a String. A String is not a primitive type like int, char, boolean, and double are. Primitive types always start with lowercase letters, but a String starts with a capital letter. This makes it an object.

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know in Java: ● Printing, Variables, & Types ● User Input ● Arithmetic Expressions ● Casting ● Logical/Comparison Operators with Booleans ● For & While Loops ● If Statements ● Loop-and-a-Half ● Short-Circuit Evaluation ● De Morgan's Laws ● Strings	Students will be able to: ● call system class methods to generate output to the console ● write programs using variables by declaring, initializing and assigning a value to their variable ● ask for user input and print out the input in the console program by using readLine, readInt, readDouble, and/or readBoolean ● construct logical statements using boolean variables and logical operators in Java ● create their own if statements to selective choose which code is executed in their programs in Java ● create programs that use the loop-and-a-half structure to repeat code until a SENTINEL is met, causing the program to break out of the loop ● identify situations where Java will use short circuit evaluation when running a program ● explain what De Morgan’s Laws are ● compare Strings correctly using .equals instead of ==

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time

-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student's 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 8

Java: Methods

Summary and Rationale

Students will learn about programming methods, return values, parameters, and writing methods that perform some advanced manipulation of Strings. This unit also examines the similarities and differences between the String class and the char primitive type.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

Computing Systems	
3A-CS-01	Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.
3A-CS-03	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
Algorithms & Programming	
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-14	Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs
3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3A-AP-22	Design and develop computational artifacts working in team roles using collaborative tools.
Impacts of Computing	
A-IC-26	Demonstrate ways a given algorithm applies to problems across disciplines.
Computer Science & Design Thinking Practices	
Recognizing and Defining Computational Problems	
Creating Computational Artifacts	
Fostering an Inclusive Computing and Design Culture	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.

9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.
9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus

Enduring Understandings:	**Essential Questions:**
Methods allow us to break down our code into reusable parts, avoid using repeated code, and simplify our code. Essentially, they are like “functions” that we explored in Python. Parameters are inputs to methods. If methods are boxes, then parameters are inputs to our boxes (what goes inside the box). Parameters allow us to generalize our methods so that they can solve more than one specific instance of a problem, and instead can solve different versions of the same problem (for example, add 10 to any given number). A method is like assigning a task. A return value is giving the programmer the completed work. This allows the programmer to write methods that do some work and send back the result. Parameters are like inputs into the function, and the return value is output. Comments are an important part of programming style. Programs are usually written by teams of people and maintained for many years. Others need to be able to read your code and understand what is going on. Javadocs, which are a standard, agreed-upon way of commenting Java code. There is a tool, also called Javadoc, that is able to read Javadoc comments and create clean, clear documentation for your Java	What is the purpose of methods? How are methods similar to structures in Python? Explain the purpose of returning a value from a method. Why is commenting your code important? What is the difference between a compile time error and a runtime error?

programs. Javadoc refers to both the tool and the style of commenting Java code.

String methods are called in order to get information about them and manipulate them to form new Strings.
Strings are simply sequences of chars. All char values have a corresponding int value. chars are actually stored as numbers! For example, 65 corresponds to ‘A’, 66 corresponds to ‘B’, and so on. Special characters like tabs, quotes, and new lines, and how to store and print these special characters using escape sequences like ‘\n’ are helpful. The Character class provides several useful methods that allow us to manipulate and get information about char values.

When there is a bug in the program, Java will actually provide helpful information about where the bug is and what kind of bug is occurring by throwing an Exception. There are different kinds of exceptions and what they mean in Java that help debug the programs.

It’s time to put methods of String classes and Character classes all together with looping through the characters of a String to write some methods that perform some advanced manipulation of Strings.

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know in Java:

- Methods, Parameters, Return Values
- Javadocs
- Strings Methods
- Strings and Characters
- Exceptions

Students will be able to:

- Use parameters to generalize their methods to solve general problems.
- Create programs that call methods with return values and store the result for later use.
- Create Javadoc comments to document their methods
- Utilize String methods to create programs that manipulate Strings in interesting ways

	● Utilize the static methods of the Character class to manipulate get information about char values ● Utilize exceptions to find and fix bugs in programs ● Implement pseudocode solutions in Java

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions
Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/

Unit 9

Java: Classes and Object-Oriented Programming

Summary and Rationale

Students will learn about classes and objects, the foundation of Object Oriented Programming. Students will be creating instances of objects and learning about the overarching idea of “using a class as a client”. Students will also learn about the basics of writing classes including implementing constructors, using instance variables, and writing a toString method, as well as getter and setter methods.

Recommended Pacing

. For recommended pacing refer to the scope and sequence for each course.

Standards

CSTA K-12 Computer Science Standards (2017)

AP - Algorithms & Programming

2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.
2-AP-17	Systematically test and refine programs using a range of test cases.
3A-AP-13	Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
3A-AP-15	Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs

3A-AP-21	Evaluate and refine computational artifacts to make them more usable and accessible.
3B-AP-12	Compare and contrast fundamental data structures and their uses.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
Computer Science & Design Thinking Practices	
Creating Computational Artifacts	
Collaborating Around Computing and Design	
Testing and Refining Computational Artifacts	
Fostering an Inclusive Computing and Design Culture	
Career Readiness, Life Literacies and Key Skills	
9.2.12.CAP.3	Investigate how continuing education contributes to one's career and personal growth.
9.2.12.CAP.4	Evaluate different careers and develop various plans.
9.2.12.CAP.8	Determine job entrance criteria (e.g., education credentials, math/writing/reading comprehension tests, drug tests) used by employers in various industry sectors.
9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2	Identify career pathways that highlight personal talents, skills, and abilities.
9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition.
9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice.
9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving.
9.4.12.DC.5	Debate laws and regulations that impact the development and use of software.
9.4.12.DC.7	Evaluate the influence of digital communities on the nature, content and responsibilities of careers, and other aspects of society.
9.4.12.DC.8	Explain how increased network connectivity and computing capabilities of everyday objects allow for innovative technological approaches to climate protection.
9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

9.4.12.TL.3	Analyze the effectiveness of the process and quality of a collaborative environment.
9.4.12.TL.4	Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem.
Interdisciplinary Connections	
Standard	
RL.10	Read and comprehend complex literary and informational texts independently and proficiently.
W.1	Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.
9.1	All students will demonstrate the creative, critical thinking, collaboration, and problem solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Instructional Focus

Enduring Understandings:	**Essential Questions:**
Objects are things that have state and behavior, and classes are the templates for creating objects. An object is something that has both state and behavior. An object is an instance of a class. A class is a template for creating objects. Objects are instantiated from classes. Objects and instances generally refer to the same thing When someone else creates a class, and you use the functionality of that class, that is called being a client of the class. Users don't need to know exactly how everything works inside the class, they can just use all the functionality of that class without worrying about the inner workings. For example, you can use the String class to create String objects, and call methods on the objects to get information like the length and the characters at certain indices, without ever looking at the source code for the String class. In Java, everything is represented as a class. Classes are a nice way of bundling up functionality. There are classes written for a lot of common needs (such as the String class and the Character class), and we can use these classes in our programs. The instance variables define the state of an object (instance), and the instance methods define the behavior of an object (instance).	What does object oriented programming mean? How is object oriented programming different from functional programming? Explain the benefit of being able to use the functionality of a class without ever having to look at the source code for the class. What is the purpose of getter and setter methods?

Instance methods give the object new abilities, they define what the object can actually do.

toString is one example of an instance method, it returns the instance's (object's) instance variables put together as a String, that way the instance can be printed out to the screen. It gives an object the ability, or the behavior, to represent itself as a String. In this lesson, we'll start adding more instance methods to our classes.

Getter and setter methods give the client access to the private instance variables in a class. Programmers don't want to make instance variables public because then the client has full access to them, and might accidentally do something that messes them up (for example, it doesn't make sense to have a negative width or a negative height for a Rectangle object). Instead, programmers expose limited access to them through getter and setter methods. That way, they can allow only certain changes to the instance variables, and block other changes.

Evidence of Learning (Assessments)

Tests
Quizzes
Homework
Programs
Projects
Class participation

Objectives (SLO)

Students will know in Java:

- Introduction to Classes and Objects
- Classes vs. Objects
- Using a Class as a Client
- Writing Classes
- Writing Classes and Instance Methods
- Getter and Setter Methods

Students will be able to:

- Create programs that create (instantiate) multiple objects from a given class and call methods on an object and print out the result
- Create multiple objects of a given class
- Call methods on an object to access the object's state and behavior
- Create programs that use other classes as a client to solve a specific problem
- Determine what instance variables a class should have and create them
- Call getter and setter methods on an object to access the object's private instance variables

Suggested Resources/Technology Tools

-Programming compiler codeHS.com
-Online notebook/journal
-Schoology

Tier 1 Modifications and Accommodations

Including special education students, Multilingual Language Learners (MLLs), students at risk of school failure, gifted and talented students, and students with 504 plans;

General Modifications for students struggling to learn:
Small group instruction within the classroom
Differentiation through content, process, product,and environment
Individual feedback and praise towards what is done correctly based upon effort, attitude and strategy.
Help students manage individual stressors for the student and plan alternate pathways for completion of assignments.

Special Education:
Modifications are determined by each student's Individual Education Plan. Examples include:
-Use concrete examples of concepts before teaching the abstract
-Reduce the number of concepts presented at one time
-Give additional presentations of program solutions by varying the methods using repetition, simpler programming structures, more examples and modeling
-Use of aids (manipulatives and program examples)
-Frequently check on progress of independent work
-Provide study guides and copy of notes
-Provide repetition and practice

MLL:
Modifications are determined by each student. Examples include:
- Provide students with notes, examples, tests, and quizzes in their primary language
- Monitor the student's comprehension of language used during instruction
- Give written directions to supplement verbal directions
- Frequently check on progress of independent work

504:
Modifications are determined by each student’s 504 plan. Examples include:
-Teacher will review, restate and repeat directions, as needed
-Frequently check on progress of independent work

Career Readiness, Life Literacies, and Key Skills NJSLS

Please select all standards that apply to this unit of study:

Act as a responsible and contributing community members and employee
Attend to financial well-being
Consider the environmental, social and economic impacts of decisions

Demonstrate creativity and innovation
Utilize critical thinking to make sense of problems and persevere in solving them
Model integrity, ethical leadership and effective management
Plan education and career paths aligned to personal goals
Use technology to enhance productivity increase collaboration and communicate effectively
Work productively in teams while using cultural/global competence

Suggestions on integrating these standards can be found at: https://www.nj.gov/education/standards/clicks/