Summit Public Schools
Summit, New Jersey
Grade Level: 8 / Content Area: Introduction to Computer Science Cycle/ 12 weeks

Overview:

Introduction to Computer Science is a cycle course that serves as an introduction to programming for students with no prior experience. Students will explore a number of foundational programming concepts using Python. Topics will include variables, data types, conditional statements, for and while loops, strings, lists and debugging strategies. Students will also develop computational thinking skills and problem-solving techniques.

Unit 1: Introduction to Programming
This unit introduces students to foundational computer science topics such as algorithms, data types, loops, conditionals and debugging strategies while exploring the history of the Python programming language.
Unit 2: Introduction to Python
This unit introduces students to the Python development environment and will be able to write their first Python programs in a terminal window.
Unit 3: Conditional Statements
In this unit, students will learn to create decision-making logic through if statements, comparison and logical operators.
Unit 4: Loops
In this unit, students will learn to develop iterative structures (for and while loops) to create more complex programs.
Unit 5: Functions and Exceptions
In this unit, students will be introduced to functions, namespace and debugging techniques.
Unit 6: Strings
In this unit, students will explore string manipulation, including indexing, slicing and concatenation.
Unit 7: Advanced Data Structures
In this unit, students will be introduced to more advanced data structures such as lists and tuples.

Unit 1: Introduction to Programming	
Big Ideas: *Course Objectives/Content Statement(s)* ● Students will learn how to break down problems into step by step instructions. ● Students will learn to use loops to repeat commands efficiently. ● Students will learn to create blocks of reusable code, functions.	
Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
● What is computer programming? ● How can we break down a problem into smaller steps? ● How does the order of instructions impact our program? ● How can we repeat parts of a program without repeating the same code? ● How can we create reusable pieces of code to improve efficiency?	Students will understand that: ● Computer programming is creating instructions for computers to solve specific tasks. ● Programs are built from smaller building blocks of code. ● The correct sequencing of instructions is essential for a program to work as intended. ● Loops and conditionals allow us to create efficient code to solve complex problems without repetition. ● Functions help us create reusable, readable and more organized programs.
Areas of Focus: Proficiencies **(New Jersey Student Learning Standards)**	**Lessons**
NJ Standard 8.1 - Computer Science • 8.1.12.IC.1: Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices. • 8.1.12.IC.3: Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources. • 8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug. **ISTE Standard 1.1. Empowered Learner:** 1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to	**Lesson 1:** ● Explain a "command" - single line of code to instruct the computer to "do something" ● Demonstrate basic Tracy functions including moving forward, drawing lines and turning. ● Students will practice basic Tracy functions. **Lesson 2:** ● Introduce For Loops ● Students will practice using for loops to cut down on repetitive code. **Lesson 3:**

transfer their knowledge to explore emerging technologies.

Career-Ready Practices
CRP1: Act as a responsible and contributing citizen and employee.
CRP2: Apply appropriate academic and technical skills.
CRP4: Communicate clearly and effectively and with reason.
CRP5: Consider the environmental, social and economic impacts of decisions.
CRP6: Demonstrate creativity and innovation.
CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11: Use technology to enhance productivity.

- Students will practice navigating Tracy through more complicated exercises, where specific angles are needed to draw to draw the shapes.

Lesson 4:
- Explain the importance of good comments as documentation to keep code organized and readable.
- Explain the importance of giving variables descriptive names.

Lesson 5:
- Introduce functions as a way to create small, reusable blocks of code
- Explain how functions save time and improve organization.
- Students will practice grouping commands into functions.

Lesson 6:
- Introduce variables which are used to store data.
- Students will practice using variables to store and use data in a program.

Lesson 7:
- Introduce using User-Input to allow users to interact with programs as they run.
- Introduce using parameters to input variables into functions.
- Students will practice using user input to create different program outcomes.

Lessons 8:
- Introduce simple looping structures and conditional statements
- Introduce the counter variable used in for loops
- Students will practice using the value of the counter variable and controlling how its value is changed.

	Lesson 9-10: ● Introduce conditional statements, including comparison operators. ● Students will practice controlling program flow through conditional statements. **Lesson 11:** ● Students will practice setting up more complex control structures using boolean statements, loops and conditional statements.
Differentiation	**Assessments**
-Self-paced lessons naturally accommodate individual learning speeds and styles. **-Specific examples will be provided for students who need more support.** **-Extension projects are offered for each project.** **-Notes are shared with the entire class.** **Interdisciplinary Connections** ● Science and Math - Hands-on application of mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles. **Technology Integration** ● A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments. **Media Literacy Integration** ● STEAM Career Webquest ● Python in the news projects **Global Perspectives** ● **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.** **Supports for English Language Learners**	**Formative Assessments:** ● **Teacher will monitor the room and conference with each student** ● **Exit tickets** ● **Partner code reviews** **Summative Assessment:** **End of unit programs enable students to demonstrate understanding and application of the Python concepts learned throughout the unit.**

Sensory Supports	Graphic Supports	Interactive Supports
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group
Magazines & Newspapers	Timelines	Structures
Physical activities	Number lines	Internet / Software support
Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**
Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response provided via computer or electronic device	Increase opportunities to engage in active academic responding	Individualized assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic	Modified assessment grading

	mapping		

Recommended Texts to Support Unit:

Unit 2: Introduction to Python	
Big Ideas: *Course Objectives/Content Statement(s)* ● Students will explore the development environment. ● Students will learn how to write and run their first Python program using the terminal window. ● Students will use variables with different data types. ● Students will create dynamic programs which react to user input.	
Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
● How can we create programs that interact with our users? ● What are the different data types that are used in Python? ● How can Python perform mathematical operations? ● How can we modify text in Python? ● What happens when a program encounters an error?	Students will understand that: ● User input allows programs to be interactive. ● Python uses data types to distinguish between different types of data (numbers, text, true/false) ● Operations perform actions on data to compare or calculate. ● String operators modify text data. ● Debugging skills help identify and fix errors in code.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	**Lessons**
NJ Standard 8.1 - Computer Science • 8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values. **ISTE Standard 1.1. Empowered Learner:** 1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. **ISTE Standard 1.5. Computational Thinker:**	**Lesson 1:** ● Introduce print statements and outputting text to the terminal window. ● Introduce data types including integer, boolean, float, and String. ● Introduce using user-input through the terminal window. ● Students will practice formatting print statements. ● Students will practice identifying the types of variables. ● Students will practice using user-input to modify variables using the terminal window.

<table>
<tr>
<td>1.5.d Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

ISTE Standard 1.6. Creative Communicator:
1.6.b Students create original works or responsibly repurpose or remix digital resources into new creations.

1.6.c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.

Career-Ready Practices
CRP1: Act as a responsible and contributing citizen and employee.
CRP2: Apply appropriate academic and technical skills.
CRP4: Communicate clearly and effectively and with reason.
CRP6: Demonstrate creativity and innovation.
CRP7: Employ valid and reliable research strategies.
CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11: Use technology to enhance productivity.
CRP12: Work productively in teams while using cultural global competence.</td>
<td>● Students will practice manipulating variables with different data types.

Lesson 2:
● Introduce syntax of mathematical operators.
● Introduce the difference between normal division and integer division.
● Students will practice manipulating variables using mathematical operators.

Lesson 3:
● Introduce concatenating Strings
● Introduce explicit type casting.
● Students will practice concatenating Strings and creating print statements that include both Strings and numbers.

Lesson 4:
● Review using print commands and using Python with a terminal window.
● Review the importance of effective comments.
● Students will practice effective comments by reviewing past programs and using comments to organize and explain code.

Lesson 5:
● Students will practice using flowcharts to help plan their programs and show their algorithmic logic.</td>
</tr>
<tr>
<th>Differentiation</th>
<th>Assessments</th>
</tr>
<tr>
<td>-Self-paced lessons naturally accommodate individual learning speeds and styles.
-Specific examples will be provided for students who need more support.
-Extension projects are offered for each project.
-Notes are shared with the entire class.

Interdisciplinary Connections
● Science and Math - Hands-on application of</td>
<td>Formative Assessments:
● Teacher will monitor the room and conference with each student
● Exit tickets
● Partner code reviews

Summative Assessment:
End of unit programs and projects enable students to demonstrate understanding and application of the</td>
</tr>
</table>

mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles.

Technology Integration

- A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments.

Media Literacy Integration

- STEAM Career Webquest
- Python in the news projects

Global Perspectives

- **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.**

Supports for English Language Learners		
Sensory Supports	**Graphic Supports**	**Interactive Supports**
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group
Magazines & Newspapers	Timelines	Structures
Physical activities	Number lines	Internet / Software support
Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Python concepts learned throughout the unit.

Sample Project: Rock-Paper-Scissors

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**
Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response provided via computer or electronic device	Increase opportunities to engage in active academic responding	Individualized assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic mapping	Modified assessment grading

Recommended Texts to Support Unit:

Unit 3: Conditional Statements

Big Ideas: *Course Objectives/Content Statement(s)*

- Students will use conditions to make program choices.
- Students will combine conditions with "and", "or" and "not" for complex decisions.
- Students will use operators to compare values and control program flow.

Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
• How can we create programs that interact with our users? • How can we write programs that behave differently depending on specified conditions? • What are the different data types that are used in Python? • How can Python perform mathematical operations? • How can we modify text in Python? • What happens when a program encounters an error?	Students will understand that: • User input allows programs to be interactive. • Python uses data types to distinguish between different types of data (numbers, text, true/false) • Comparison operators perform actions on data to compare values. • Logical operators can be used to write statements that check multiple conditions. • String operators modify text data. • Debugging skills help identify and fix errors in code.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	**Lessons**
NJ Standard 8.1 - Computer Science 8.1.8.DA.1: Organize and transform data collected using computational tools to make it usable for a specific purpose. 8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values. **ISTE Standard 1.1. Empowered Learner:** 1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.	**Lesson 1-2:** • Introduce boolean variables and how to use them in conditional statements. • Review writing conditional statements (if/else) using comparison operators. • Students will practice using booleans and comparison operators to write conditional statements that impact program flow. **Lesson 3:** • Students will practice using booleans with If Statements. **Lesson 4:**

<table>
<tr>
<td>

ISTE Standard 1.5. Computational Thinker:
1.5.d Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

ISTE Standard 1.6. Creative Communicator:
1.6.b Students create original works or responsibly repurpose or remix digital resources into new creations.

1.6.c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.

Career-Ready Practices
CRP1: Act as a responsible and contributing citizen and employee.
CRP2: Apply appropriate academic and technical skills.
CRP4: Communicate clearly and effectively and with reason.
CRP5: Consider the environmental, social and economic impacts of decisions.
CRP6: Demonstrate creativity and innovation.
CRP7: Employ valid and reliable research strategies.
CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11: Use technology to enhance productivity.
CRP12: Work productively in teams while using cultural global competence.

</td>
<td>

- Introduce comparison operators which allow students to add complexity to their programs.
- Students will practice using comparison operators to check if numbers are greater than, less than or equal to other numbers and control the flow of their programs.

Lesson 5:
- Introduce logical operators which enable students to test multiple conditionals at once.
- Students will practice using logical operators with programs that will react differently if a condition is not true.

Lesson 6:
- Introduce floating point numbers and how to round them in Python.
- Students will practice using floats in programs while reviewing booleans, comparison and logical operators.

</td>
</tr>
<tr>
<th>Differentiation</th>
<th>Assessments</th>
</tr>
<tr>
<td>

-Self-paced lessons naturally accommodate individual learning speeds and styles.
-Specific examples will be provided for students who need more support.
-Extension projects are offered for each project.
-Notes are shared with the entire class.

Interdisciplinary Connections
- Science and Math - Hands-on application of mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles.

Technology Integration

</td>
<td>

Formative Assessments:
- **Teacher will monitor the room and conference with each student**
- **Exit tickets**
- **Partner code reviews**

Summative Assessment:
End of unit programs and projects enable students to demonstrate understanding and application of the Python concepts learned throughout the unit.

Sample Project: Choose Your Own Adventure Story

</td>
</tr>
</table>

- A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments.

Media Literacy Integration

- STEAM Career Webquest
- Python in the news projects

Global Perspectives

- **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.**

Supports for English Language Learners		
Sensory Supports	**Graphic Supports**	**Interactive Supports**
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group
Magazines & Newspapers	Timelines	Structures
Physical activities	Number lines	Internet / Software support
Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**

(include creating a flowchart)

Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response provided via computer or electronic device	Increase opportunities to engage in active academic responding	Individualized assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic mapping	Modified assessment grading

Recommended Texts to Support Unit:

Unit 4: Loops

Big Ideas: *Course Objectives/Content Statement(s)*

- Students will use loops to repeat sections of code efficiently
- Students will use loops to automate repetitive tasks.
- Students will be able to nest loops when necessary.

Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
• How can we repeat blocks of code efficiently using loops? • How do conditions control when a loop continues or stops? • How do break and continue statements allow us to better control the flow of our programs? • How do nested loops allow us to create more complete programs?	Students will understand that: • Loops are an effective way to reduce code repetition and improve code readability. • while loops execute code repeatedly as long as a condition remains True. • while loops can become infinite loops if the condition never becomes False. • for loops iterate a specific number of times based on a counter variable. • for-each loops iterate over a sequence of values, executing code for each item. • break and continue statements provide options to exit a loop early or skip specific iterations. • A loop-and-a-half structure can be used to intentionally create an infinite loop that ends when a condition is met. • Nesting loops and combining loops enables us to code complex program logic.
Areas of Focus: Proficiencies **(New Jersey Student Learning Standards)**	**Lessons**
NJ Standard 8.1 - Computer Science 8.1.8.DA.1: Organize and transform data collected using computational tools to make it usable for a specific purpose. 8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values.	**Lesson 1:** • Introduce While Loops • Introduce how infinite loops are created and how to avoid them • Students will practice using while loops to continue running blocks of code until a condition is false.

8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. 8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users. 8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug. **ISTE Standard 1.1. Empowered Learner:** 1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. **ISTE Standard 1.5. Computational Thinker:** 1.5.d Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions. **ISTE Standard 1.6. Creative Communicator:** 1.6.b Students create original works or responsibly repurpose or remix digital resources into new creations. 1.6.c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. **Career-Ready Practices** **CRP1**: Act as a responsible and contributing citizen and employee. **CRP2**: Apply appropriate academic and technical skills. **CRP4**: Communicate clearly and effectively and with reason. **CRP6**: Demonstrate creativity and innovation. **CRP8**: Utilize critical thinking to make sense of problems and persevere in solving them. **CRP11**: Use technology to enhance productivity. **CRP12**: Work productively in teams while using cultural global competence.	**Lesson 2:** ● Introduce For Loops to create more complex programs. ● Students will practice using for loops which increment in different ways. ● Students will practice using for loops to accumulate data (numbers and Strings) **Lesson 3:** ● Introduce Break and Continue statements. ● Introduce loop-and-a-half structure. ● Students will practice using break and continue statements to allow them more control over the program flow. **Lesson 4:** ● Introduce nesting loops and if-statements ● Students will practice using nested loops and conditional statements to write more complex programs. **Lesson 5:** ● Students will practice using pseudocode to construct conditional statements and more complex programs.
Differentiation	**Assessments**

-Self-paced lessons naturally accommodate individual learning speeds and styles.
-Specific examples will be provided for students who need more support.
-Extension projects are offered for each project.
-Notes are shared with the entire class.

Interdisciplinary Connections

- Science and Math - Hands-on application of mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles.

Technology Integration

- A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments.

Media Literacy Integration

- STEAM Career Webquest
- Python in the news projects

Global Perspectives

- **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.**

Supports for English Language Learners		
Sensory Supports	**Graphic Supports**	**Interactive Supports**
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group
Magazines & Newspapers	Timelines	Structures

Formative Assessments:

- **Teacher will monitor the room and conference with each student**
- **Exit tickets**
- **Partner code reviews**

Summative Assessment:
End of unit programs and projects enable students to demonstrate understanding and application of the Python concepts learned throughout the unit.

Sample Project:
Lottery Game Project (include creating pseudocode)

Physical activities	Number lines	Internet / Software support
Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**
Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response provided via computer or electronic device	Increase opportunities to engage in active academic responding	Individualized assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic mapping	Modified assessment grading

Recommended Texts to Support Unit:

Unit 5: Functions and Exceptions	
Big Ideas: *Course Objectives/Content Statement(s)* ● Students will use functions to group instructions together. ● Students will use parameters to make functions accept and use different data. ● Students will understand and use variable scope properly. ● Students will create forward thinking error handling to create robust code.	
Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
● How can we write code that is reusable and efficient? ● How can we use functions as abstractions to write more readable code? ● How can functions accept different inputs to perform different tasks? ● What is variable scope? ● How can functions output values that can be used elsewhere in a program? ● How can we write programs that anticipate and handle errors effectively?	Students will understand that: ● Functions improve code organization and efficiency. ● Functions should be used to represent tasks that must be performed multiple times in a program. ● When writing programs that solve complex problems, functions can be used to organize subproblems. ● Parameters help create functions that can perform multiple tasks based on different inputs. ● Variable scope impacts which variables can be used and can impact program behavior. ● Functions can output return values that can be stored and used elsewhere in programs. ● Error handling needs to be integrated into program design so that programs are robust.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	**Lessons**
NJ Standard 8.1 - Computer Science 8.1.8.DA.1: Organize and transform data collected using computational tools to make it usable for a specific purpose. 8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their	**Lesson 1:** ● Review functions without parameters. ● Students will practice grouping commands together for efficiency. ● Review parameters, inputs to functions.

values.

8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.

8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.

ISTE Standard 1.1. Empowered Learner:
1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

ISTE Standard 1.5. Computational Thinker:
1.5.d Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

ISTE Standard 1.6. Creative Communicator:
1.6.b Students create original works or responsibly repurpose or remix digital resources into new creations.

1.6.c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations.

Career-Ready Practices
CRP1: Act as a responsible and contributing citizen and employee.
CRP2: Apply appropriate academic and technical skills.
CRP4: Communicate clearly and effectively and with reason.
CRP6: Demonstrate creativity and innovation.
CRP8: Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11: Use technology to enhance productivity.
CRP12: Work productively in teams while using cultural global competence.

- Students will practice using functions with different inputs to create more flexible and efficient programs

Lesson 2:
- Introduce namespace and scope of variables.
- Students will practice understanding and using variables with different scope.**:**
- Students will debug code snippets to show understanding of using variables with different scope.

Lesson 3:
- Introduce functions which return values.
- Students will practice using functions that accept and return different values.

Lesson 4:
- Students will extend functionality of existing functions by rewriting them to return values.

Lesson 5:
- Introduce exceptions and error handling.
- Students will practice using error messages and debugging using try/except statements.

Lesson 6:
- Students will practice using error messages with code completion and code correction exercises.

Lesson 7:
- Students will review all of the concepts introduced in this unit including functions, parameters, namespace, return values, exceptions and error handling.

Differentiation

-Self-paced lessons naturally accommodate individual learning speeds and styles.
-Specific examples will be provided for students who need more support.
-Extension projects are offered for each project.
-Notes are shared with the entire class.

Interdisciplinary Connections

- Science and Math - Hands-on application of mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles.

Technology Integration

- A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments.

Media Literacy Integration

- STEAM Career Webquest
- Python in the news projects

Global Perspectives

- **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.**

Supports for English Language Learners

Sensory Supports	Graphic Supports	Interactive Supports
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group

Assessments

Formative Assessments:

- **Teacher will monitor the room and conference with each student**
- **Exit tickets**
- **Partner code reviews**

Summative Assessment:
End of unit programs and projects enable students to demonstrate understanding and application of the Python concepts learned throughout the unit.

Sample Project: Calculator

Magazines & Newspapers	Timelines	Structures
Physical activities	Number lines	Internet / Software support
Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**
Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response provided via computer or electronic device	Increase opportunities to engage in active academic responding	Individualized assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic mapping	Modified assessment grading

Recommended Texts to Support Unit:

Unit 6: Strings	
Big Ideas: *Course Objectives/Content Statement(s)* ● Students will learn best practices for using strings in Python. ● Students will be able to manipulate strings using indexes, slicing and concatenation. ● Students will use built-in string methods.	
Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
● How can we store and represent text data within Python programs? ● How can we access and modify specific characters within a string? ● How can we rearrange and combine strings? ● How do the built-in string methods affect our Python programs?	Students will understand: ● How to define and create strings to represent text data. ● How to use indexing to access and modify individual characters within a string. ● How to use slicing and concatenation to manipulate the order of string characters. ● How to use for and for-each loops to iterate through the characters in a string. ● Built-in strong methods help create readable and efficient code while searching, replacing and extracting characters from strings.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	**Lessons**
NJ Standard 8.1 - Computer Science 8.1.8.DA.1: Organize and transform data collected using computational tools to make it usable for a specific purpose. 8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values. 8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	**Lesson 1-2:** ● Introduce strings and indexes. ● Introduce slicing strings to obtain substrings. ● Introduce immutability of strings. ● Students will practice using indexes to identify each character in a string. ● Students will practice using slicing and concatenating strings to rearrange character order within strings. ● Students will practice modifying strings showing understanding of immutability.

8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users. 8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug. **ISTE Standard 1.1. Empowered Learner:** 1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. **ISTE Standard 1.5. Computational Thinker:** 1.5.d Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions. **ISTE Standard 1.6. Creative Communicator:** 1.6.b Students create original works or responsibly repurpose or remix digital resources into new creations. 1.6.c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. **Career-Ready Practices** **CRP1**: Act as a responsible and contributing citizen and employee. **CRP2**: Apply appropriate academic and technical skills. **CRP4**: Communicate clearly and effectively and with reason. **CRP6**: Demonstrate creativity and innovation. **CRP8**: Utilize critical thinking to make sense of problems and persevere in solving them. **CRP11**: Use technology to enhance productivity. **CRP12**: Work productively in teams while using cultural global competence.	**Lesson 3:** ● Introduce iterating through the indexes in a String using for loops and the len function. ● Introduce for-each loops to iterate through each character in a String. ● Students will practice using loops to iterate through Strings. **Lesson 4:** ● Introduce "in" keyword to check if a String contains a specific substring. ● Introduce built-in string methods. ● Students will practice writing conditions using the "in" keyword. ● Students will practice using Python's built-in string methods. **Lesson 5:** ● Students will review all of the concepts introduced in this unit including string indexes, slicing, concatenation, immutability and built-in string methods.
Differentiation	**Assessments**
-Self-paced lessons naturally accommodate individual learning speeds and styles. **-Specific examples will be provided for students who need more support.**	**Formative Assessments:** ● **Teacher will monitor the room and conference with each student** ● **Exit tickets**

-Extension projects are offered for each project.
-Notes are shared with the entire class.

Interdisciplinary Connections

- Science and Math - Hands-on application of mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles.

Technology Integration

- A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments.

Media Literacy Integration

- STEAM Career Webquest
- Python in the news projects

Global Perspectives

- **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.**

Supports for English Language Learners		
Sensory Supports	**Graphic Supports**	**Interactive Supports**
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group
Magazines & Newspapers	Timelines	Structures
Physical activities	Number lines	Internet / Software support

- **Partner code reviews**

Summative Assessment:
End of unit programs and projects enable students to demonstrate understanding and application of the Python concepts learned throughout the unit.

Sample Project: Palindromes or Pig Latin

Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**
Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response provided via computer or electronic device	Increase opportunities to engage in active academic responding	Individualized assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic mapping	Modified assessment grading

Recommended Texts to Support Unit:

Unit 7: Advanced Data Structures	
Big Ideas: *Course Objectives/Content Statement(s)* ● Students will learn how to use tuples and lists to store collections of data. ● Students will learn to use both tuples and lists though built-in methods and functions. ● Students will learn how immutability of tuples can be useful when data shouldn't be modified. ● Students will learn how mutability of lists can be useful for dynamic data storage.	
Essential Questions *What provocative questions will foster inquiry, understanding, and transfer of learning?*	**Enduring Understandings** *What will students understand about the big ideas?*
● How can we store collections of data in Python programs? ● How can we access and manipulate data stored in a list? ● What built-in methods and functions are available for working with tuples and lists?	Students will understand: ● How to create and use lists to store and manage dynamic data sets. ● How to use for and for-each loops to iterate through the elements of lists. ● How to use built-in methods to access elements of lists efficiently.
Areas of Focus: Proficiencies (New Jersey Student Learning Standards)	**Lessons**
NJ Standard 8.1 - Computer Science 8.1.8.DA.1: Organize and transform data collected using computational tools to make it usable for a specific purpose. **ISTE Standard 1.1. Empowered Learner:** 1.1.d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies. **ISTE Standard 1.5. Computational Thinker:** 1.5.d Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions. **ISTE Standard 1.6. Creative Communicator:**	**Lesson 1:** ● Introduce lists and how they can be used to manage program complexity ● Review indexes and explain that lists use the same indexing as Strings. **Lesson 2:** ● Students will practice using lists to manage program complexity, as well as iterating through them. **Lesson 3:** ● Introduce built-in methods/functions associated with lists. ● Students will practice using the built-in methods and functions associated with lists.

1.6.b Students create original works or responsibly repurpose or remix digital resources into new creations. 1.6.c Students communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. **Career-Ready Practices** **CRP1**: Act as a responsible and contributing citizen and employee. **CRP2**: Apply appropriate academic and technical skills. **CRP4**: Communicate clearly and effectively and with reason. **CRP6**: Demonstrate creativity and innovation. **CRP8**: Utilize critical thinking to make sense of problems and persevere in solving them. **CRP11**: Use technology to enhance productivity. **CRP12**: Work productively in teams while using cultural global competence.	
Differentiation	**Assessments**
-Self-paced lessons naturally accommodate individual learning speeds and styles. **-Specific examples will be provided for students who need more support.** **-Extension projects are offered for each project.** **-Notes are shared with the entire class.** **Interdisciplinary Connections** ● Science and Math - Hands-on application of mathematical concepts such as logic, data organization and problem solving and iterative engineering design principles. **Technology Integration** ● A LMS designed specifically for programming will be used to monitor rosters, data and coding assignments. **Media Literacy Integration** ● STEAM Career Webquest ● Python in the news projects **Global Perspectives**	**Formative Assessments:** ● **Teacher will monitor the room and conference with each student** ● **Exit tickets** ● **Partner code reviews** **Summative Assessment:** **End of unit programs and projects enable students to demonstrate understanding and application of the Python concepts learned throughout the unit.**

- **We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used making the user part of an international community.**

Supports for English Language Learners		
Sensory Supports	**Graphic Supports**	**Interactive Supports**
Real-life objects	Charts	In pairs or partners
Manipulatives	Graphic Organizers	In triands or small groups
Pictures	Tables	In a whole group
Illustrations, diagrams & drawings	Graphs	Using cooperative group
Magazines & Newspapers	Timelines	Structures
Physical activities	Number lines	Internet / Software support
Videos & Film		In the home language
Broadcasts		With mentors
Models & Figures		

Intervention Strategies		
Accommodations	**Interventions**	**Modifications**
Allow for verbal responses	Multi-sensory techniques	Modified tasks/expectations
Repeat/confirm directions	Increase task structure (e.g. directions, checks for understanding, feedback	Differentiated materials
Permit response	Increase	Individualized

provided via computer or electronic device	opportunities to engage in active academic responding	assessment tools based on student need
Audio Books	Utilize pre-reading strategies and activities previews, anticipatory guides, and semantic mapping	Modified assessment grading

Recommended Texts to Support Unit: