Unit 3: Solving with Syntax

Content Area: **Template**
Course(s):
Time Period: **Full Year**
Length: **Full Year**
Status: **Published**

UNIT RATIONALE

The goal of Unit 3 is for students to begin to understand and use the flexibility and power of programming in a text-based environment. Students will be introduced to the Python® programming language in the collaborative Cloud9 development environment. In this unit, students will continue to build on coding fundamentals as they apply the same coding concepts, computational thinking practices, and development processes introduced in units 1 and 2.

ESSENTIAL QUESTIONS

How am I applying independent, cooperative, and collaborative strategies to find my own answers?

Why is computer science considered a form of art and creative expression?

What are some essential operations you do over and over with lists or collections?

How does parameterization generalize a specific solution?

Why is it important to become a creator of apps and not just a user?

STANDARDS

NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

New Jersey (NJSLS) - Grades 9-12 - Computer Science and Design Thinking (2020)

8.1.12.CS.1:
Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.3:
Compare the functions of application software, system software, and hardware.
8.1.12.CS.4:
Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
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.

8.1.12.DA.2:

Describe the trade-offs in how and where data is organized and stored.

8.1.12.AP.1:

Design algorithms to solve computational problems using a combination of original and existing algorithms.

8.1.12.AP.2:

Create generalized computational solutions using collections instead of repeatedly using simple variables.

8.1.12.AP.3:

Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

8.1.12.AP.4:

Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

8.1.12.AP.5:

Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

8.1.12.AP.6:

Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

8.1.12.AP.7:

Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.

8.1.12.AP.8:

Evaluate and refine computational artifacts to make them more usable and accessible.

8.1.12.AP.9:

Collaboratively document and present design decisions in the development of complex programs.

8.2.12.ED.5:

Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
CS.9-12.8.1.12.AP.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
CS.9-12.8.1.12.AP.7	Collaboratively design and develop programs and artifacts for broad audiences by

	incorporating feedback from users.
CS.9-12.8.1.12.AP.8	Evaluate and refine computational artifacts to make them more usable and accessible.
CS.9-12.8.1.12.AP.9	Collaboratively document and present design decisions in the development of complex programs.
CS.9-12.8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
CS.9-12.8.1.12.CS.3	Compare the functions of application software, system software, and hardware.
CS.9-12.8.1.12.CS.4	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
CS.9-12.8.1.12.DA.2	Describe the trade-offs in how and where data is organized and stored.
CS.9-12.8.1.12.IC.1	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
CS.9-12.8.1.12.IC.3	Predict the potential impacts and implications of emerging technologies on larger social, economic, and political structures, using evidence from credible sources.
CS.9-12.8.2.12.ED.5	Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).

NEW JERSEY STUDENT LEARNING STANDARDS: CAREER READINESS, LIFE LITERACIES AND KEY SKILLS

12.9.3.ST.2	Use technology to acquire, manipulate, analyze and report data.
12.9.3.IT-PRG.2	Demonstrate the use of industry standard strategies and project planning to meet customer specifications.
12.9.3.IT-PRG.3	Analyze system and software requirements to ensure maximum operating efficiency.
12.9.3.IT-PRG.4	Demonstrate the effective use of software development tools to develop software applications.
12.9.3.IT-PRG.5	Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
12.9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.
12.9.3.IT-PRG.8	Perform quality assurance tasks as part of the software development cycle.
12.9.3.IT-PRG.9	Perform software maintenance and customer support functions.
12.9.3.IT-PRG.10	Design, create and maintain a database.
12.9.3.MN-QA.4	Employ project management processes using data and tools to deliver quality, value-added products.
12.9.3.ST-ET.2	Display and communicate STEM information.
12.9.3.ST-ET.3	Apply processes and concepts for the use of technological tools in STEM.

NEW JERSEY STUDENT LEARNING STANDARDS: COMPUTER SCIENCE AND DESIGN THINKING

See content area standards.

PRE-ASSESSMENTS

Write a short program to make sure you (the student) can get input from a user and show output on the screen.

INSTRUCTIONAL PLAN

MODULE 1

LESSON 3.1

Lesson 3.1 Text-Based Coding

In this lesson, students will reinforce previously learned concepts as they are introduced to the power of programming in a text-based language. The goal of this lesson is for students to become comfortable implementing algorithms using conditionals and loops in Python.

Activity 3.1.1 Python Programing (1 day)

Activity 3.1.2 Variables and Conditionals (3 days)

Project 3.1.3 Combo Menu (5 days)

Activity 3.1.1

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Activity 3.1.1** Get started with Python®. Apply coding fundamentals in a text-based language. Apply file naming conventions and version control.
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions

Success Criteria	From a given section of code, identify the function, the function call, the variable, the strings, and data types.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will practice the same coding fundamentals they learned earlier in the course, but instead of switching between block programming and python, they will use Python 3 as their primary programming language.
Suggested Modifications	**English Language Learners** **Adjusted Speech**: The teacher changes speech patterns to increase student comprehension. This could include facing the students, paraphrasing, clearly indicating the most important ideas, and speaking more slowly. **Visuals**: The teacher uses graphics, pictures, visuals, and manipulatives. This helps ELL students better understand and comprehend the subjects at hand. **Front-Loading Vocabulary**: The teacher front loads vocabulary This means providing students with a list of important vocabulary words they will need to know for a book, lesson, etc. prior to the lesson being taught. Including pictures to go with the vocabulary words is also very beneficial for the students. **Students with Individualized Education Plans/504s** **Chunking**: The teacher presents information in a way that makes it easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded by excessive detail. The best way to deliver information is to organize it into meaningful units. Because students with special needs get overloaded easily, chunking is an effective strategy to use with them. **Checking for Understanding**: It is important to constantly chec for understanding, especially for students who have accommodations. Teachers want to make sure students understand the concepts being covered in a way that makes sense to them.

Extra time: The teacher provides students with special needs extra time to complete work or answer questions. It is important to give students enough time to process their thoughts.

Oral Reading: The teacher will read work orally to students. Class work such as tests and literature circles may need to be read aloud to the student.

Gifted & Talented Strategies

Extensions/Enrichments: Teachers will provide gifted and talented students with extension/enrichment projects. Students will be challenged to further their understanding, to apply acquired knowledge, and/or to produce something in reference t acquired knowledge.

Modify/Change Activities: Teachers will monitor and modify activities to accommodate those students who need to be challenged further. Additional reading, problem-solving, writing, or project work is necessary for those students who are ready to move on at a rate more accelerated than their peers. In this way G & T students are provided the same opportunity for support as special needs students.

Students at Risk of School Failure

Directions or Instructions: Make sure directions and/or instructions are given in limited numbers. Give directions/instructions verbally and in simple written format. Ask students to repeat the instructions or directions to ensure understanding occurs. Check back with the student to ensure he/she hasn't forgotten.

Peer Support: Peers can help build confidence in other students by assisting in peer learning. Many teachers use the 'ask 3 before me' approach. This is fine, however, a student at risk may have to have a specific student or two to ask. Set this up for the student so he/she knows who to ask for clarification before going to you.

Alternate or Modified Assignments: Always ask yourself, "How can I modify this assignment to ensure the students at risk are able to complete it?" Sometimes you'll simplify the task, reduce the length of the assignment or allow for a different mode of delivery. For instance, many students may hand something in, the at-risk student may jot notes and give you the information

	verbally. Or, it just may be that you will need to assign an alternate assignment. **Increase One to One Time:** When other students are working, always touch base with your students at risk and find out if they're on track or needing some additional support. A few minutes here and there will go a long way to intervene as the need presents itself. **Contracts:** It helps to have a working contract between you and your students at risk. This helps prioritize the tasks that need to be done and ensure completion happens. Each day write down what needs to be completed, as the tasks are done, provide a checkmark or happy face. The goal of using contracts is to eventually have the student come to you for completion sign-offs **Hands On:** As much as possible, think in concrete terms and provide hands-on tasks. This means a child doing math may require a calculator or counters. The child may need to tape record comprehension activities instead of writing them. A child may have to listen to a story being read instead of reading it him/herself. **Tests/Assessments:** Tests can be done orally if need be. Breal tests down in smaller increments by having a portion of the test in the morning, another portion after lunch and the final part the next day. **Seating:** Seat students near a helping peer or with quick access to the teacher. Those with hearing or sight issues need to be close to the instruction which often means near the front.

Activity 3.1.2

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Activity 3.1.2** Apply coding fundamentals in a text-based language Develop and test code incrementally Develop a program independently
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions

Success Criteria	The features of the program were created correctly.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will use a small subset of the Python functions to experiment with familiar coding concepts,
Suggested Modifications	See Activity 3.1.1

Project 3.1.3

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Project 3.1.3** Apply coding fundamentals in a text-based language Apply file naming conventions and version control Develop and test code incrementally Develop a program independently
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Working meal ordering program.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will create a program that automates the ordering of a meal by offering options in the menu, capturing the responses, an providing a summary of the order to the user.
Suggested Modifications	See Activity 3.1.1

LESSON 3.2

Lesson 3.2 Text-based Solutions

In this lesson, students will continue to explore the use of text-based programming. The lesson ends with students creating a game simulation that allows them to make generalizations and develop functions that attempt to detect and react to another team's strategy.

Activity 3.2.1 Lists and Elements (4 days)

Activity 3.2.2 Social Media: Posting Objects (6 days)

Activity 3.2.3 Iteration and Counts (5 days)

Activity 3.2.4 Course Registration: Functions (7 days)

Project 3.2.5 Artificial Intelligence: Rock, Paper, Scissors Simulation (8 days)

Project 3.2.5

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Project 3.2.5** Decompose a project into smaller parts Apply coding fundamentals and iterative processes Develop a simulation as part of a Scrum team
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Students are able to participate in the Rock, Paper, Scissors tournament.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Develop a program that attempts to predict the outcome of a seemingly random game—rock-paper-scissors. Then students wi compete to see which team has the best algorithm (most wins).
Suggested Modifications	See Activity 3.1.1

Activity 3.2.4

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Activity 3.2.4** Functions allow a programmer to set aside an entire section of co and tell the program to execute that section from top to bottom whenever a given set of syntax known as a function call is encountered. Once the set-aside section of code is fully executed flow of control returns to the point of the function call.
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Students are able to create and call their own function that displa the text "Assignment 1 Completed"
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Modify a course registration program to explore functions in Pytho
Suggested Modifications	See Activity 3.1.1

Activity 3.2.3

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Activity 3.2.3** Use for and while loops to manage program flow.
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Students' programs asks the user for orders repeatedly (iteratively using loops) until the user enters 'q'.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will modify their previously developed code to accept mo than one order.
Suggested Modifications	See Activity 3.1.1

Activity 3.2.2

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Activity 3.2.2** Create, manipulate, and access lists in Python Pair program to develop a program that solves a problem and generates understanding
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	The features of the program work correctly.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will modify a program that posts messages and gets pos to understand how social media works.
Suggested Modifications	See Activity 3.1.1

Activity 3.2.1

Student Learning Intentions (SLI) WALT: (We are learning to…)	**Activity 3.2.1** Learn to create, manipulate, and access lists in Python® Pair program to develop a program that solves a problem and generates understanding
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration

	Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	When the user is done entering input, the program should inform them of their menu selections, for only the items they ordered, and the total cost of the order.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will modify the Combo Menu program from the previous activity to create a list for each order placed and store it to be accessed later.
Suggested Modifications	See Activity 3.1.1

INTERDISCIPLINARY CONNECTIONS: NEW JERSEY STUDENT LEARNING STANDARDS FOR ELA, SOCIAL STUDIES, SCIENCE AND/OR MATHEMATICS

CCSS - English-Language Arts

Key Ideas and Details:

CCSS.ELA-LITERACY.RL.11-12.1 Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.

Integration of Knowledge and Ideas:

CCSS.ELA-LITERACY.W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

Production and Distribution of Writing:

CCSS.ELA-LITERACY.W.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Research to Build and Present Knowledge:

CCSS.ELA-LITERACY.W.11-12.7 Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.

Range of Writing:

CCSS.ELA-LITERACY.W.11-12.10 Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences

CCSS - Mathematics

Reason quantitatively and use units to solve problems:

CCSS.MATH.CONTENT.HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.

Create equations that describe numbers or relationships:

CCSS.MATH.CONTENT.HSA-CED.A.1 Create equations and inequalities in one variable and use them to solve problems.

Analyze functions using different representations:

CCSS.MATH.CONTENT.HSF-IF.C.7 Graph functions expressed symbolically and show key features of the graph.

Apply geometric concepts in modeling situations:

CCSS.MATH.CONTENT.HSG-MG.A.1 Use geometric shapes, their measures, and their properties to describe objects

Calculate expected values and use them to solve problems:

CCSS.MATH.CONTENT.HSS-MD.A.1 Define a random variable for a quantity of interest by assigning a numerical value to each event in a sample space;

CCSS.MATH.CONTENT.HSS-MD.A.2 Calculate the expected value of a random variable;

English Language Arts

- Journal writing
- Close reading of industry-related content
- Create a brochure for a specific industry
- Keep a running word wall of industry vocabulary

Social Studies

- Research the history of a given industry/profession
- Research prominent historical individuals in a given industry/profession
- Use historical references to solve problems

World Language

- Translate industry-content
- Create a translated index of industry vocabulary
- Generate a translated list of words and phrases related to information technology

Math

- Compare and contrast use of equations and variables in algebra and programming.
- Program graphics and use the properties of geometric shapes
- Compare the computer graphic coordinate system with the Cartesian coordinate plane in math
- Compare probability and the use of random numbers in computer programming.
- Track and track various data, such as industry's impact on the GDP, career opportunities or among of individuals currently occupying careers

Fine & Performing Arts

- Create a poster recruiting young people to focus their studies on a career in Information Technology

Science

- Research the environmental impact of a given career or industry
- Research latest developments in Information technology
- Investigate applicable-careers in STEM fields

REFLECTIONS