Unit 1: Variables and Functions

Content Area: **Math**
Course(s): **Generic Course, WOOD I**
Time Period: **Marking Period 1**
Length: **4 weeks**
Status: **Published**

Standards

Math Processing Standards

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

Life Literacies and Key Skills

TECH.9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
TECH.9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

Computer Science Standards

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

Transfer Goals

Transfer Goals

Students will be able to independently use their learning to solve problems with a real-world context using variables, expressions, statements, and functions.

Concepts

Essential Questions

- How does one think like a computer scientist?
- In what ways do functions allow for a more readable and useful program?
- What are some methods for debugging a program?
- What decisions must be made when declaring and assigning variables?

Understandings

Students will understand that...

- A function is a named sequence of statements that performs a computation.
- Although it can be frustrating, debugging is one of the most intellectually rich, challenging, and interesting parts of programming.
- It is a good idea to add notes to your programs to explain in natural language what the program is doing.
- One of the most powerful features of a programming language is the ability to manipulate variables.
- Programming languages are formal languages that have been designed to express computations.
- Programs are sequences of instructions that specify how to perform a computation.
- Three of the major differences between natural and formal languages are ambiguity, redundancy, and literalness.
- When reading a program, it often makes more sense to follow the flow of execution rather than reading from top to bottom.

Critical Knowledge and Skills

Knowledge

Students will know:

- A function definition specifies the name of a new function and the sequence of statements that runs when the function is called.
- A statement is a unit of code that has an effect, like creating a variable or displaying a value.
- An assignment statement creates a new variables and gives it a value.
- An expression is a combination of values, variables, and operators.
- Integers, floating-point numbers, and strings are all types of values in Python.
- Many functions are passed variables called parameters when called.
- The Python operators for addition, subtraction, multiplication, division, and exponentiation.
- The Python operators for string concatenation and repetition.
- The difference between fruitful and void functions.
- The order of operations used in Python programming.
- The process of determining the structure of a statement is called parsing.

Skills

Students will be able to:

- Combine and adapt previously created functions to achieve a new goal.
- Create a variable and assign it a value.
- Create and use state and stack diagrams.
- Debug a simple program.
- Describe why functions are necessary for a concise, useful program.
- Parse lines of code.
- Use the Python interpreter as a calculator.
- Write a function to execute a basic sequence of commands.
- Write a simple program directly in the console.
- Write comments to clarify the purpose of a program.

Assessment and Resources

School Formative Assessment Plan (Other Evidence)

- Chapter Exercises
- Debugging Exercises
- Notebook
- Open Note Quizzes

School Summative Assessment Pan

- Unit Assessment (done via LinkIt)
- Unit Programming Project

Primary Resources

- *Think Python: How to Think Like a Computer Scientist, 2nd ed.*
  - http://greenteapress.com/thinkpython2/html/index.html
  - Chapters 1-3

- PythonAnywhere Interpreter
  - http://www.pythonanywhere.com

Supplementary Resources

- Runestone Academy: *How to Think Like a Computer Scientist, Interactive Edition*
  - https://runestone.academy/runestone/static/thinkcspy/index.htm

- The Python Standard Library
  - https://docs.python.org/3/library/index.html

Technology Integration and Differentiated Instruction

Technology Integration

- **Python Anywhere**
  - This online interpreter allows students to write and run Python3 programs in the browser. Students are also able to save files and collaborate inside the suite.

- **Runestone Academy**
  - This interactive version of the textbook provides some further examples and videos. It also

have several exercises that are automatically scored, allowing for immediate feedback.

- **Google Products**
  - Google Classroom - Used for daily interactions with the students covering a vast majority of different educational resources (Daily Notes, Exit Tickets, Classroom Polls, Quick Checks, Additional Resources/ Support, Homework, etc.)
  - GAFE (Google Apps For Education) - Using various programs connected with Google to collaborate within the district, co-teachers, grade level partner teacher, and with students to stay connected with the content that is covered within the topic. Used to collect data in real time and see results upon completion of the assignments to allow for 21st century learning.

- **One to One Student laptop**
  - All students within the West Deptford School District are given a computer, allowing for 21st century learning to occur within every lesson/topic.

NJCCCS -- Technology 9-12

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.2	Analyze the relationships between internal and external computer components.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Differentiated Instruction

- **Gifted Students (N.J.A.C.6A:8-3.1)**
  - Within each lesson, the Gifted Students are given choice on topic and subject matter allowing them to explore interests appropriate to their abilities, areas of interest and other courses.

- **English Language Learners (N.J.A.C.6A:15)**
  - Within each lesson, the English Language Learners are given choice of topic and resources so that their materials are within their ability to grasp the language.
  - All assignments have been created in the student's native language.
  - Work with ELL Teacher to allow for all assignments to be completed with extra time.

- **At-Risk Students (N.J.A.C.6A:8-4.3c)**
  - Within each lesson, the at-risk students are given choice of topic and resources so that their

materials are within their ability level and high-interest.

- **Special Education Students (N.J.A.C.6A:8-3.1)**
  - Within each lesson, special education students are given choice of topic and resources so that their materials are within their ability level and high-interest.
  - All content will be modeled with examples and all essays are built on a step-by-step basis so modifications for assignments in small chunks are met.
  - All other IEP modifications will be honored (ie. hard copies of notes, directions restated, etc.)

Interdisciplinary Connections

LANGUAGE ARTS

- Students will explore and exlain the difference between formal and natural languages.

SCIENCE

- Students will explore the Scientific Method when writing programs.

SOCIAL STUDIES

- Students will consider the idea that people naturally respond to computers as if they are people.

WORLD LANGUAGES

VISUAL/PERFORMING ARTS

- Students will use functions to create geometric patterns and designs.

BUSINESS EDUCATION

GLOBAL AWARENESS

Learning Plan / Pacing Guide

WEEK 1

- Setup Google Classroom, PythonAnywhere accounts
- Complete Chapter 1 Notes
- Chapter 1 Exercises
- Begin Chapter 2 Notes

WEEK 2

- Complete Chapter 2 Notes
- Chapter 2 Exercises

- Open Note Quiz
- Begin Chapter 3 Notes

WEEK 3

- Complete Chapter 3 Notes
- Chapter 3 Exercises
- Begin Unit Programming Project

WEEK 4

- Complete Unit Programming Project
- Unit Asssessment

Unit 2: Conditionals and Recursion

Content Area: **Math**
Course(s): **Generic Course, WOOD I**
Time Period: **Marking Period 1**
Length: **4 weeks**
Status: **Published**

Standards

Math Processing Standards

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

Life Literacies and Key Skills

TECH.9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
TECH.9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
TECH.9.4.12.TL.3	Analyze the effectiveness of the process and quality of collaborative environments.

Computer Science Standards

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.

Transfer Goals

Transfer Goals

Students will be able to independently use their learning to solve problems with a real-world context using repetition, encapsulation, conditional execution, and recursion.

Concepts

Essential Questions

- How are conditional statements used to enhance a program?
- How are repetition structures used to enhance a program?
- How can functions work together to add functionality to a program?
- What are some methods for debugging a program?

Understandings

Students will understand that:

- Conditional statements allow actions to take place only when a specific condition is met, or behave differently based on parameter value.
- Conditional statements allow programs to make decisions about which lines of code run and which do not.
- Recursion is the process of executing a function which calls itself.
- Refactoring is a way to improve interfaces and facilitate code re-use.
- Using parameters for generalization allows functions to be useful in many more situations.

Critical Knowledge and Skills

Knowledge

Students will know:

- A function that calls itself is said to be recursive.
- Adding a parameter to a function is called generalization.
- Infinite recursion occurs when a recursive function never reaches a base case.
- Methods are similar to functions, but have a slightly different syntax.
- Often, the most important parts of a syntax or runtime error are the type of error and the line on which it occurred.
- Preconditions and postconditions can help with debugging.
- The utility of 'if' and 'if-else' statements, chained and nested conditionals.
- The utility of floor division, modulus operator, relational operators, and logical operators.
- Wrapping a piece of code in a function is called encapsulation.

Skills

Students will be able to:

- Draw a Koch curve using the turtle module.
- Draw a stack diagram for a generalized function.
- Write a function that checks mathematical theorems such as Fermat's Last Theorem or the Triangle Inequality Theorem.
- Write a script that reads the current epoch time and converts it to a time of day in hours, minutes, and seconds.

Assessment and Resources

School Formative Assessment Plan (Other Evidence)

- Chapter Exercises
- Debugging Exercises
- Notebook

- Open Note Quizzes

School Summative Assessment Pan

- Unit Assessment (done via LinkIt)
- Unit Programming Project

Primary Resources

- *Think Python: How to Think Like a Computer Scientist, 2nd ed.*
  - http://greenteapress.com/thinkpython2/html/index.html
  - Chapters 4-5

- PythonAnywhere Interpreter
  - http://www.pythonanywhere.com

- Trinket.io
  - https://trinket.io/turtle

Supplementary Resources

- Runestone Academy: *How to Think Like a Computer Scientist, Interactive Edition*
  - https://runestone.academy/runestone/static/thinkcspy/index.htm

- The Python Standard Library
  - https://docs.python.org/3/library/index.html

Technology Integration and Differentiated Instruction

Technology Integration

- **Python Anywhere**
  - This online interpreter allows students to write and run Python3 programs in the browser. Students are also able to save files and collaborate inside the suite.

- **Trinket.io**
  - This online interpreter allows for use of the Tkinter GUI toolkit (which is not available in PythonAnywhere as it is a server side environment.) This is necessary for the use of the turtle module as part of the interface design case study.

- **Runestone Academy**
  - This interactive version of the textbook provides some further examples and videos. It also have several exercises that are automatically scored, allowing for immediate feedback.

- **Google Products**
  - Google Classroom - Used for daily interactions with the students covering a vast majority of different educational resources (Daily Notes, Exit Tickets, Classroom Polls, Quick Checks, Additional Resources/ Support, Homework, etc.)
  - GAFE (Google Apps For Education) - Using various programs connected with Google to collaborate within the district, co-teachers, grade level partner teacher, and with students to stay connected with the content that is covered within the topic. Used to collect data in real time and see results upon completion of the assignments to allow for 21st century learning.

- **One to One Student laptop**
  - All students within the West Deptford School District are given a computer, allowing for 21st century learning to occur within every lesson/topic.

NJCCCS -- Technology 9-12

TECH.8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
TECH.8.1.12.B.CS2	Create original works as a means of personal or group expression.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.2	Analyze the relationships between internal and external computer components.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Differentiated Instruction

- **Gifted Students (N.J.A.C.6A:8-3.1)**
  - Within each lesson, the Gifted Students are given choice on topic and subject matter allowing them to explore interests appropriate to their abilities, areas of interest and other courses.

- **English Language Learners (N.J.A.C.6A:15)**
  - Within each lesson, the English Language Learners are given choice of topic and resources so that their materials are within their ability to grasp the language.

- All assignments have been created in the student's native language.
- Work with ELL Teacher to allow for all assignments to be completed with extra time.

- **At-Risk Students (N.J.A.C.6A:8-4.3c)**
  - Within each lesson, the at-risk students are given choice of topic and resources so that their materials are within their ability level and high-interest.

- **Special Education Students (N.J.A.C.6A:8-3.1)**
  - Within each lesson, special education students are given choice of topic and resources so that their materials are within their ability level and high-interest.
  - All content will be modeled with examples and all essays are built on a step-by-step basis so modifications for assignments in small chunks are met.
  - All other IEP modifications will be honored (ie. hard copies of notes, directions restated, etc.)

Interdisciplinary Connections

LANGUAGE ARTS

- Students will understand the need for appropriate syntax.
- Students will read technical documentation, including error reports.
- Students will write a development plan.

SCIENCE

- Students will explore the Scientific Method when writing programs.

SOCIAL STUDIES

- Students will reflect on the changes in the worlds of art and mathematics brought about by programming innovation.

WORLD LANGUAGES

VISUAL/PERFORMING ARTS

- Students will use turtle art to create different shapes, including Koch curves.

BUSINESS EDUCATION

GLOBAL AWARENESS

Learning Plan / Pacing Guide

WEEK 1

- Begin Chapter 4 Notes

- Turtle Module Exercises
- Continue Chapter 4 Notes

WEEK 2

- Complete Chapter 4 Notes
- Chapter 4 Exercises
- Open Note Quiz
- Begin Chapter 5 Notes

WEEK 3

- Complete Chapter 5 Notes
- Chapter 5 Exercises
- Begin Unit Programming Project

WEEK 4

- Complete Unit Programming Project
- Unit Asssessment

Unit 3: Fruitful Functions and Iteration

Content Area: **Math**
Course(s): **Generic Course, WOOD I**
Time Period: **Marking Period 2**
Length: **4 weeks**
Status: **Published**

Standards

Math Processing Standards

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

Life Literacies and Key Skills

TECH.9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
TECH.9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).

Computer Science Standards

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.

Transfer Goals

Transfer Goals

Students will be able to independently use their learning to solve problems with a real-world context using functions, composition, iteration (including while loops and break statements), and reassignment.

Concepts

Essential Questions

- How are algorithms related to computer programming?
- How does incremental development help programmers to handle increasingly complex programs?
- What are some methods for debugging a program?
- What are some of the benefits and potential harmful effects of reassignment?
- Why is it important to ensure functions use type checking for its arguments?

Understandings

Students will understand that:

- Designing algorithms is interesting, intellectually challenging, and a central part of computer science.
- Fruitful functions are functions that return a value.
- Often, when you come to function call, instead of following the flow of execution,one assumes that the function works correctly and return the right result.
- Python (specifically, the subset of Python we've learned thus far!) is a complete programming language, which means that anything that can be computed can be expressed in this language.
- Reassigning variables is often useful, but should be used with caution. If the values of variables change

frequently, it can make the code difficult to read and debug.

- Repeating identical or similar tasks without making errors is something computers do well and people do poorly.
- The goal of incremental development is to avoid long debugging sessions by adding and testing only a small amount of code at a time.

Critical Knowledge and Skills

Knowledge

Students will know:

- A guardian is a programming pattern that uses a conditional statement to check for and handle circumstances that might cause an error.
- A temporary variable is used to store an intermediate value in a complex calculation.
- An algorithm is a general process for solving a category of problems.
- An infinite loop is a loop in which the terminating condition is never satisfied.
- Incremental development is a program development plan intended to avoid debugging by adding and testing only a small amount of code at a time.
- Iteration is the repeated execution of a set of statements using either a recursive function call or a loop.
- Scaffolding refers to code that is used during program development but is not part of the final version.
- To increment refers to an update that increases the value of a variable (often by one).

Skills

Students will be able to:

- Combine two functions by composition.
- Debug a program by breaking it down into smaller pieces.
- Describe a process in terms of an algorithm.
- Use and describe the process of incremental development.
- Use reassignment in a function.
- Use type checking in their functions.
- Write a fruitful function.
- Write a while loop and describe why it is the appropriate choice for the function.

Assessment and Resources

School Formative Assessment Plan (Other Evidence)

- Chapter Exercises
- Debugging Exercises
- Notebook
- Open Note Quizzes

School Summative Assessment Pan

- Unit Assessment (done via LinkIt)
- Unit Programming Project

Primary Resources

- *Think Python: How to Think Like a Computer Scientist, 2nd ed.*
    - http://greenteapress.com/thinkpython2/html/index.html
    - Chapters 6-7

- PythonAnywhere Interpreter
    - http://www.pythonanywhere.com

- Trinket.io
    - https://trinket.io/turtle

Supplementary Resources

- Runestone Academy: *How to Think Like a Computer Scientist, Interactive Edition*
    - https://runestone.academy/runestone/static/thinkcspy/index.htm

- The Python Standard Library
    - https://docs.python.org/3/library/index.html

Technology Integration and Differentiated Instruction

Technology Integration

- **Python Anywhere**
  - This online interpreter allows students to write and run Python3 programs in the browser. Students are also able to save files and collaborate inside the suite.

- **Trinket.io**
  - This online interpreter allows for use of the Tkinter GUI toolkit (which is not available in PythonAnywhere as it is a server side environment.) This is necessary for the use of the turtle module as part of the interface design case study.

- **Runestone Academy**
  - This interactive version of the textbook provides some further examples and videos. It also have several exercises that are automatically scored, allowing for immediate feedback.

- **Google Products**
  - Google Classroom - Used for daily interactions with the students covering a vast majority of different educational resources (Daily Notes, Exit Tickets, Classroom Polls, Quick Checks, Additional Resources/ Support, Homework, etc.)
  - GAFE (Google Apps For Education) - Using various programs connected with Google to collaborate within the district, co-teachers, grade level partner teacher, and with students to stay connected with the content that is covered within the topic. Used to collect data in real time and see results upon completion of the assignments to allow for 21st century learning.

- **One to One Student laptop**
  - All students within the West Deptford School District are given a computer, allowing for 21st century learning to occur within every lesson/topic.

NJCCCS -- Technology 9-12

TECH.8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
TECH.8.1.12.B.CS2	Create original works as a means of personal or group expression.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.2	Analyze the relationships between internal and external computer components.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and

engineering.

Differentiated Instruction

- **Gifted Students (N.J.A.C.6A:8-3.1)**
  - Within each lesson, the Gifted Students are given choice on topic and subject matter allowing them to explore interests appropriate to their abilities, areas of interest and other courses.

- **English Language Learners (N.J.A.C.6A:15)**
  - Within each lesson, the English Language Learners are given choice of topic and resources so that their materials are within their ability to grasp the language.
  - All assignments have been created in the student's native language.
  - Work with ELL Teacher to allow for all assignments to be completed with extra time.

- **At-Risk Students (N.J.A.C.6A:8-4.3c)**
  - Within each lesson, the at-risk students are given choice of topic and resources so that their materials are within their ability level and high-interest.

- **Special Education Students (N.J.A.C.6A:8-3.1)**
  - Within each lesson, special education students are given choice of topic and resources so that their materials are within their ability level and high-interest.
  - All content will be modeled with examples and all essays are built on a step-by-step basis so modifications for assignments in small chunks are met.
  - All other IEP modifications will be honored (ie. hard copies of notes, directions restated, etc.)

Interdisciplinary Connections

LANGUAGE ARTS

- Students will understand the need for appropriate syntax.
- Students will read technical documentation, including error reports.
- Students will write a program that evaluates palindromes.

SCIENCE

- Students will explore the Scientific Method when writing programs.
- Students will read and write algorithms to describe processes.

SOCIAL STUDIES

- Students will consider different algorithms for evaluating pi throughout history.

WORLD LANGUAGES

- Students will consider the manner in which different cultures created different algorithms for evaluating pi throughout history.

VISUAL/PERFORMING ARTS

BUSINESS EDUCATION

- Students will write a program with outputs a table based on certain inputs, much like a basic spreadsheet.

GLOBAL AWARENESS

- Students will consider the manner in which different cultures created different algorithms for evaluating pi throughout history.

Learning Plan / Pacing Guide

WEEK 1

- Begin Chapter 6 Notes
- Debugging Interlude
- Continue Chapter 6 Notes

WEEK 2

- Complete Chapter 6 Notes
- Chapter 6 Exercises
- Open Note Quiz
- Begin Chapter 7 Notes

WEEK 3

- Complete Chapter 7 Notes
- Chapter 7 Exercises
- Begin Unit Programming Project

WEEK 4

- Complete Unit Programming Project
- Unit Asssessment

Unit 4: Strings and Lists

Content Area: **Math**
Course(s): **Generic Course, WOOD I**
Time Period: **Marking Period 2**
Length: **4 weeks**
Status: **Published**

Standards

Math Processing Standards

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

Life Literacies and Key Skills

TECH.9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
TECH.9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
TECH.9.4.12.TL.1	Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
TECH.9.4.12.TL.3	Analyze the effectiveness of the process and quality of collaborative environments.

Computer Science Standards

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.

Transfer Goals

Transfer Goals

Students will be able to independently use their learning to solve problems with a real-world context using strings and lists.

Concepts

Essential Questions

- How can lists and list methods be used to improve functionality?
- How can strings and string methods be used to improve program functionality?
- What are some methods for debugging a program?

Understandings

Students will understand that:

- A string is a sequence, meaning it is an ordered collection of other values -- specifically characters.
- A list is a sequence whose values can be of any type.
- Lists are mutable.
- Lists have operators and methods which allow them to be sorted and changed.
- String methods allow programmers to process large text files, including searching.

- Strings are immutable.

Critical Knowledge and Skills

Knowledge

Students will know:

- A delimiter is a character or string used to indicate where a string should be split.
- A nested list is a list that is an element of another list.
- A sequence is an ordered collection of values where each value is identified by an integer index.
- A special case is a test case which is atypical or non-obvious, and often cause errors in a program.
- Aliasing is a circumstance where two or more variables refer to the same object.
- An accumulator is a variable used in a loop to add up or accumulate a result.
- An index is an integer value used to select an item in a sequence.
- In Python, indices start from 0.
- Reduce, map, and filter are sequence processing patterns that traverse sequences or lists.
- To traverse is to iterate through the items in a sequence, performing a similar operation on each.

Skills

Students will be able to:

- Read and search through a .txt file in a Python console.
- Compare two strings.
- Create a traversal with either a for or while loop.
- Map, filter, and reduce a list.
- Use the list methods 'append', 'extend', and 'sort'.
- Use the string methods 'upper', 'lower', 'find', and 'in'.
- Write a function that counts the occurrences of a value is found in a string or a list.

Assessment and Resources

School Formative Assessment Plan (Other Evidence)

- Chapter Exercises
- Debugging Exercises
- Notebook
- Open Note Quizzes

School Summative Assessment Pan

- Unit Assessment (done via LinkIt)
- Unit Programming Project

Primary Resources

- *Think Python: How to Think Like a Computer Scientist, 2nd ed.*
    - http://greenteapress.com/thinkpython2/html/index.html
    - Chapters 8-10

- PythonAnywhere Interpreter
    - http://www.pythonanywhere.com

- Trinket.io
    - https://trinket.io/turtle

Supplementary Resources

- Runestone Academy: *How to Think Like a Computer Scientist, Interactive Edition*
    - https://runestone.academy/runestone/static/thinkcspy/index.htm

- The Python Standard Library
    - https://docs.python.org/3/library/index.html

Technology Integration and Differentiated Instruction

Technology Integration

- **Python Anywhere**
    - This online interpreter allows students to write and run Python3 programs in the browser.

Students are also able to save files and collaborate inside the suite.

- **Trinket.io**
  - This online interpreter allows for use of the Tkinter GUI toolkit (which is not available in PythonAnywhere as it is a server side environment.) This is necessary for the use of the turtle module as part of the interface design case study.

- **Runestone Academy**
  - This interactive version of the textbook provides some further examples and videos. It also have several exercises that are automatically scored, allowing for immediate feedback.

- **Google Products**
  - Google Classroom - Used for daily interactions with the students covering a vast majority of different educational resources (Daily Notes, Exit Tickets, Classroom Polls, Quick Checks, Additional Resources/ Support, Homework, etc.)
  - GAFE (Google Apps For Education) - Using various programs connected with Google to collaborate within the district, co-teachers, grade level partner teacher, and with students to stay connected with the content that is covered within the topic. Used to collect data in real time and see results upon completion of the assignments to allow for 21st century learning.

- **One to One Student laptop**
  - All students within the West Deptford School District are given a computer, allowing for 21st century learning to occur within every lesson/topic.

NJCCCS -- Technology 9-12

TECH.8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
TECH.8.1.12.B.CS2	Create original works as a means of personal or group expression.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.2	Analyze the relationships between internal and external computer components.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Differentiated Instruction

- **Gifted Students (N.J.A.C.6A:8-3.1)**
    - Within each lesson, the Gifted Students are given choice on topic and subject matter allowing them to explore interests appropriate to their abilities, areas of interest and other courses.

- **English Language Learners (N.J.A.C.6A:15)**
    - Within each lesson, the English Language Learners are given choice of topic and resources so that their materials are within their ability to grasp the language.
    - All assignments have been created in the student's native language.
    - Work with ELL Teacher to allow for all assignments to be completed with extra time.

- **At-Risk Students (N.J.A.C.6A:8-4.3c)**
    - Within each lesson, the at-risk students are given choice of topic and resources so that their materials are within their ability level and high-interest.

- **Special Education Students (N.J.A.C.6A:8-3.1)**
    - Within each lesson, special education students are given choice of topic and resources so that their materials are within their ability level and high-interest.
    - All content will be modeled with examples and all essays are built on a step-by-step basis so modifications for assignments in small chunks are met.
    - All other IEP modifications will be honored (ie. hard copies of notes, directions restated, etc.)

Interdisciplinary Connections

LANGUAGE ARTS

- Students will understand the need for appropriate syntax.
- Students will read technical documentation, including error reports.
- Students will read through and manipulate a .txt file which contains every word used in *Moby Dick*.

SCIENCE

- Students will explore the Scientific Method when writing programs.
- Students will read and write algorithms to describe processes.

SOCIAL STUDIES

- Students will write a program that uses the Caesar cypher, after reading about how the cypher was named for, but probably not invented by, Julius Caesar.

WORLD LANGUAGES

VISUAL/PERFORMING ARTS

BUSINESS EDUCATION

GLOBAL AWARENESS

Learning Plan / Pacing Guide

WEEK 1

- Begin Chapter 8 Notes
- Complete Chapter 8 Notes
- Chapter 8 Exercises

WEEK 2

- Begin Chapter 9 Notes (Case Study: Word Play)
- Chapter 9 Exercies (Part 1)
- Complete Chapter 9 Notes
- Chapter 9 Exercises (Part 2)

WEEK 3

- Open Note Quiz
- Begin Chapter 10 Notes
- Complete Chapter 10 Notes
- Chapter 10 Exercises

WEEK 4

- Complete Unit Programming Project
- Unit Asssessment