Randolph Township Schools
Randolph High School

Programming with Python and Java Honors

"Mathematics provides a framework for dealing precisely with notions of 'what is'. Computation provides a framework for dealing precisely with notions of 'how to'."

-- Margaret Mead

Department of
Science, Technology, Engineering, and Math
Anthony Emmons, Supervisor

Curriculum Committee:
Matthew Horner

Curriculum Developed:
June 2018

Date of Board Approval:
September 4th, 2018

Randolph Township Schools
Randolph High School

Table of Contents

Randolph Township Schools

Mission Statement

We commit to inspiring and empowering all students in Randolph schools to reach their full potential as unique, responsible and educated members of a global society.

Randolph Township Schools
Affirmative Action Statement

Equality and Equity in Curriculum

The Randolph Township School district ensures that the district's curriculum and instruction are aligned to the state's standards. The curriculum provides equity in instruction, educational programs and provides all students the opportunity to interact positively with others regardless of race, creed, color, national origin, ancestry, age, marital status, affectional or sexual orientation, gender, religion, disability or socioeconomic status.

N.J.A.C. 6A:7-1.7(b): Section 504, Rehabilitation Act of 1973; N.J.S.A. 10:5; Title IX, Education Amendments of 1972

RANDOLPH TOWNSHIP BOARD OF EDUCATION
EDUCATIONAL GOALS
VALUES IN EDUCATION

The statements represent the beliefs and values regarding our educational system. Education is the key to self-actualization, which is realized through achievement and self-respect. We believe our entire system must not only represent these values, but also demonstrate them in all that we do as a school system.

We believe:

- The needs of the child come first
- Mutual respect and trust are the cornerstones of a learning community
- The learning community consists of students, educators, parents, administrators, educational support personnel, the community and Board of Education members
- A successful learning community communicates honestly and openly in a non-threatening environment
- Members of our learning community have different needs at different times. There is openness to the challenge of meeting those needs in professional and supportive ways
- Assessment of professionals (i.e., educators, administrators and educational support personnel) is a dynamic process that requires review and revision based on evolving research, practices and experiences
- Development of desired capabilities comes in stages and is achieved through hard work, reflection and ongoing growth

Randolph Township Schools
Department of Science, Technology, Engineering, and Math

Programming with Python and Java Honors

This course is a continuation of the AP Computer Science Principles course and explores some of the more intricate aspects of computer programming. In the first semester students are introduced to programming humanoid robots and the Python programming language. Using Python, the first semester of the course emphasizes the analysis of problems, the careful selection of an appropriate algorithm, and the implementation of the algorithm in Python. Topics covered include input/output commands, control statements, looping, subroutines/functions, Python data structure, string processing, and an introduction to object-oriented programming. As further preparation for continued studies in computer science, Java is introduced in the second semester of the course. The course covers the basics of the Java language and object-oriented concepts using Finch robots for hands-on application, and Greenfoot software for game development applications. In addition to learning the structure of Java programs and the syntax of the control structures in Java, students study string and array manipulation and further explore object-oriented programming concepts, such as inheritance and polymorphism.

The application of the following Career Readiness Practices will be utilized throughout the course.

CRP1-Career Ready Practices: All students will act as a responsible and contributing citizen and employee.

CRP2-Career Ready Practices: All students will apply appropriate academic and technology skills.

CRP4-Career Ready Practices: All students will communicate clearly and effectively and with reason.

CRP5-Career Ready Practices: All students will consider the environmental, social and economic impacts of decisions.

CRP6-Career Ready Practices: All students will demonstrate creativity and innovation.

CRP8-Career Ready Practices: All students will utilize critical thinking to make sense of problems and persevere in solving them.

CRP9-Career Ready Practices: All students will model integrity, ethical leadership and effective management.

CRP10-Career Ready Practices: All students will plan education and career paths aligned to personal goals.

CRP11-Career Ready Practices: All students will use technology to enhance productivity.

CRP12-Career Ready Practices: All students will work productively in teams while using global competence.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Curriculum Pacing Chart
Programming with Python and Java Honors

SUGGESTED TIME ALLOTMENT	UNIT NUMBER	CONTENT - UNIT OF STUDY
First Semester – Python programming with application		
3 weeks	**I**	**An Introduction to Programming Robots with NAO**
3 weeks	**II**	**Introduction and Python Basics**
6 weeks	**III**	**Python Program Implementation**
4 weeks	**IV**	**Python Data Structures and Strings**
2 weeks	**V**	**Python Classes and Introduction to Object-Oriented Programming**
Second Semester – Java programming with application		
3 weeks	**VI**	**Object-Oriented Programming in Java**
6 weeks	**VII**	**Java Basics and Selection Control Structures**
7 weeks	**VIII**	**Repetition Control Structures, Strings, and Arrays**
2 weeks	**IX**	**Java Applications Projects (Finch Robot and Greenfoot)**

36 weeks is the average

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT I: An Introduction to Programming Robots with NAO

<table>
<tr><td rowspan="10">STANDARDS / GOALS:

Computer Science Teachers Association K-12 Computer Science Standards (CSTA):
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

3A-AP-17 – Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

NJSLS – Technology:
8.2.12.E.1 – Demonstrate an understanding of the problem-solving capacity of computers in our world.

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).

8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

NJSLS – 21st Century Life and Career Skills:</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Humanoid robots are complex devices with intricate software.</td><td>• Why is it necessary to understand the mechanical and programming functions of a humanoid robot?</td></tr>
<tr><td>Humans program robots to read, interpret, and interact with their surroundings.</td><td>• How can I customize how a robot interacts with its environment?</td></tr>
<tr><td>Complicated functions are comprised of simpler coding mechanisms.</td><td>• How can I break complicated functions into small tasks to make them easier to program?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Brief history of robotics and robot applications.

Hardware and software components of the robot.

Choreographe is the software used to program the NAO robot.</td><td>Students will be able to:

Identify early forms of robots and inventors in the field of robotics.

Explore the benefits of robotics in other fields of study.

Charge, switch on, and handle the robot safely.

Identify the various mechanical and electrical components of the NAO robot.

Connect the NAO robot with Choreographe software on the computer.

Utilize Choreographe drag and drop environment to enable robot to speak.</td></tr>
</table>

9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.IT-PRG.7 – Demonstrate software testing procedures to ensure quality products. 9.3.ST.6 – Demonstrate technical skills needed in a chosen STEM field. 9.3.ST-ET.3 – Apply processes and concepts for the use of technological tools in STEM. 9.3.ST-ET.4 – Apply the elements of the design process. 9.3.ST-SM.1 – Apply science and mathematics to provide results, answers and algorithms for engineering and technological activities.	**Vocabulary**: humanoid robot, box list, say box, sliders, pitch, stiffness, speech recognition	Program robot to speak, change word speed, and alter pitch. Program robot to stand, walk in the (x, y) coordinate plane, then sit. Program robot to turn and walk to a specified point. Program robot to recognize words or names. Program robot to change eye color based on user request and voice recognition.

ASSESSMENT EVIDENCE: Students will show their learning by:

- Generating robot motion and speech with block-based code http://www.aldebaran.com/en/robotics-solutions/robot-software.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students explore the capabilities of the NAO robot and develop a goal to strive for using built-in coding blocks.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT I: An Introduction to Programming Robots with NAO

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
3 Weeks	Unit 1: An Introduction to Programming Robots with NAO 1) Evolution of Robotics 2) Introduction to the NAO Robot a. Safety and setup b. Choreographe Software 3) Programming the NAO a. Speaking b. Walking c. Word recognition d. Culminating Project	Beiter, Mike, Brian Coltin, Somchaya Liemhetcharat. *An Introduction to Robotics with NAO*, 1st Edition, Aldebaran Robotics, 2013. Choreographe Software used to Program NAO Robot: http://www.aldebaran.com/en/robotics-solutions/robot-software

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT II: Introduction and Python Basics

<table>
<tr><th>STANDARDS / GOALS:</th><th>ENDURING UNDERSTANDINGS</th><th>ESSENTIAL QUESTIONS</th></tr>
<tr><td rowspan="5">CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 – Create clearly named variables that represent different data types and perform operations on their values.

3A-DA-09 – Translate between different bit representations of real-world phenomena, such as characters, numbers, and images.

NJSLS – Technology:
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).

8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

NJSLS – 21st Century Life and Career Skills:
9.3.IT-PRG.6 – Program a computer application using the appropriate computer language.

9.3.IT-PRG.7 – Demonstrate software testing procedures to ensure quality products.</td><td>Programs store data in variables and algorithms are used to manipulate that data.</td><td>• What is the difference between an iterative and a recursive algorithm and how should each be used?</td></tr>
<tr><td>Using the proper programming style enables others to more-easily understand and modify programs.</td><td>• Why are comments, indentation, and naming conventions important to the readability of a program?</td></tr>
<tr><td>An understanding of data types is essential for accurately tracking the operation of a program.</td><td>• Why is it important to know the data type of a variable?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

The IDLE Interactive Development Environment (IDE) allows writing and running of python code.

Basic Python scripts have top-down execution, where code at the top is the starting point.

The Python programming keywords and style conventions.</td><td>Students will be able to:

Download and explore the IDLE IDE.

Utilize the IDE to write simple programs.

Use pseudocode, flowcharts, or words to describe the process for changing a tire or cooking a turkey.

Describe the procedures and subprocedures needed to perform complicated tasks.

Understand the top-down execution of basic Python scripts.

Write simple programs utilizing correct style and naming conventions.</td></tr>
</table>

9.3.ST.6 – Demonstrate technical skills needed in a chosen STEM field. 9.3.ST-ET.3 – Apply processes and concepts for the use of technological tools in STEM. 9.3.ST-ET.4 – Apply the elements of the design process. 9.3.ST-SM.1 – Apply science and mathematics to provide results, answers and algorithms for engineering and technological activities.	Output formatting procedures to control how data displays on the screen. Variables must be declared and initialized. **Vocabulary**: algorithm, flowcharts, pseudocode, procedure, class, object, method, compiler, interpreter, executable, operators, constants, escape sequence, binary, unary, associativity, output manipulators, format flags, reference variables, storage allocation	Write programming code to display output to the screen and format it using the format manipulators including: • Escape sequences • string.format() Write programming code declaring and initializing variables and using them to do simple arithmetic. Write programming code to convert between data types. Write programming code to verify how much storage space is set aside for each data type.

ASSESSMENT EVIDENCE: Students will show their learning by:

- Formatting numeric output to a set number of decimal places and alignment.
- Generating simple text-based "pictures" using escape characters for special circumstances.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students download the IDLE development environment and generate their first non-block-based program to say "Hello World!".

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT II: Introduction and Python Basics

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
3 Weeks	Unit 2: Introduction and Python Basics 1) Getting Started a. Class procedures, Blackboard b. IDLE IDE c. Introduction to programming d. Top-down execution e. print() function f. Programming style 2) Data Types, Declarations, Displays a. Data constants b. Arithmetic operators c. Numerical output d. Variables and declarations e. Type conversions f. Storage size	Python and Integrated Development Environment: https://www.python.org/getit/ Python 3.6.6rc1 Documentation: https://docs.python.org/3/index.html Online Python Coding Practice: http://codingbat.com/python

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT III: Python Program Implementation

<table>
<tr><th>STANDARDS / GOALS:</th><th>ENDURING UNDERSTANDINGS</th><th>ESSENTIAL QUESTIONS</th></tr>
<tr><td rowspan="6">CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-12 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 – Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17 – Systematically test and refine programs using a range of test cases.

2-AP-19 – Document programs in order to make them easier to follow, test, and debug.

3A-AP-17 – Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-21 – Evaluate and refine computational artifacts to make them more usable and accessible.</td><td>Computer programs accept input and provide output in order to communicate with users.</td><td>• How does a programmer utilize user input to alter the flow of control in a program?</td></tr>
<tr><td>The manner in which computer programs process data can be customized by using selection and repetition structures.</td><td>• How can a program be designed to give a reasonable response, regardless of what the user enters?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Arithmetic expressions are evaluated according to order of operations, precedence rules, and the associativity of the operators used in an expression.

Built-in Math module functions exist for more advanced mathematical operations.

Many tasks require user input to the computer.

Relational and logical operators and tests are used to compare operands and evaluate as True or False.</td><td>Students will be able to:

In written and programming assignments, demonstrate an understanding of the order of operations, precedence rules, and associativity of arithmetic operators in expressions and be able to evaluate these expressions.

Write programming code that properly applies the Math module functions.

Write programming code that uses variables and constants to perform mathematical functions and evaluate arithmetic expressions.

Write programs that accept user input using the input() function and provide user output using the print() function.</td></tr>
</table>

3B-AP-16 – Demonstrate code reuse by creating programming solutions using libraries and APIs. 3B-AP-21 – Develop and use a series of test cases to verify that a program performs according to its design specifications. **NJSLS – Technology:** 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). 8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements). **NJSLS – 21st Century Life and Career Skills:** 9.2.12.C.3 – Identify transferable career skills and design alternate career paths. 9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.ST.2 – Use technology to acquire, manipulate, analyze and report data. 9.3.ST-ET.4 – Apply the elements of the design process. 9.3.ST-SM.1 – Apply science and mathematics to provide results, answers and algorithms for engineering and technological activities.	Programs can use the value of expressions to select a sequence of one or more instructions using selection statements. Programs can solve problems that require repeated calculations using repetition structures.	Distinguish between input types using the eval() function. Write programming code that correctly implements relational operators and special types of testing, including: • <, >, <=, >=, == • is • is not • isinstance() Write programming code that implements logical operators and compound statements recognizing the precedence and associativity. Write programming code that utilizes the values of "True" relational expressions (1) and "False" relational expressions (0). Write programming code that utilizes the following selection statements: • If statements • Elif statements • Nested if statements • Switch statements Write programming code that utilizes repetition structures including: • While statement • While statement with repeated user input • While statement using sentinels, break, and continue statements • For statements • Nested for loops

	User-defined functions are used to perform customized tasks within a program.	Write programming code for user-defined functions that: • Have no return value (return None) • Return a value Write programs that declare (prototype), call, and define user-written functions. Write programs that include functions that receive and pass: • Positional arguments • Keyword arguments
	A variable will have different scope (life and accessibility) depending on how and where it is declared in a program.	Write programs that utilize variables with local, enclosing function, global, and built-in scope. Use the LEGB rule to define the hierarchy level of a variable name when it is used.
	Vocabulary: assignment statement, module functions, relational operators, precedence order, selection statements, compound statements, nested if statements, switch statement, while, break, continue, for statement, function, argument, function call, return, positional arguments, keyword arguments, variable scope, LEGB rule, local variable, global variable	

ASSESSMENT EVIDENCE: Students will show their learning by:

- Creating a simple higher/lower guessing game.
- Writing a program that determines the results of the quadratic formula.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students use conditional statements to determine if an input birthdate affords the user driving privileges.
- Accept a list of names for further processing until certain exit conditions are reached.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT III: Python Program Implementation

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
6 Weeks	Unit 3: Python Program Implementation 1) Assignment and Interactive Input a. Assignment statements b. Math module functions c. input 2) Selection and Control a. Relational expressions b. elif statements c. Nested if statements d. Switch statements 3) Repetition and Control a. while statement b. Sentinels, break, continue c. for statement d. Nested loops 4) Writing Your Own Functions a. Function and argument declarations b. Returning values c. Positional and keyword arguments d. Variable scope	Python and Integrated Development Environment: https://www.python.org/getit/ Python 3.6.6rc1 Documentation: https://docs.python.org/3/index.html Online Python Coding Practice: http://codingbat.com/python

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT IV: Python Data Structures and Strings

<table>
<tr><th>STANDARDS / GOALS:</th><th>ENDURING UNDERSTANDINGS</th><th>ESSENTIAL QUESTIONS</th></tr>
<tr><td rowspan="4">CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 – Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17 – Systematically test and refine programs using a range of test cases.

3A-DA-10 – Evaluate the tradeoffs in how data elements are organized and where data is stored.

3A-AP-14 – Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-15 – Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.</td><td>Arrays are storage structures in programming languages that help manage large quantities of data.</td><td>• How does an array help a programmer deal with large amounts of data?</td></tr>
<tr><td>The input, manipulation, and output of text is very important in computer programming for interfacing with the user.</td><td>• Why is it essential that programs read and write text?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Arrays are data structures that can store a list of values of the same data type.

Array elements are stored in contiguous locations in memory and when arrays are passed as arguments the address of the array is passed, not the entire array of values.</td><td>Students will be able to:

Write programming code that declares and initializes:
• One-dimensional arrays
• Two-dimensional arrays

Write a function that can retrieve the values from an array and use these to perform various mathematical functions, e.g., traversing an array and calculating the average of the values contained.

Write programming code that passes the name of an array as an argument.

Write programming code that verifies the changes made to an array that is passed to a function are actually stored in the original array.</td></tr>
</table>

3A-AP-21 – Evaluate and refine computational artifacts to make them more usable and accessible.
 	
3B-AP-12 – Compare and contrast fundamental data structures and their uses.
 	
NJSLS – Technology:
8.2.12.E.1 – Demonstrate an understanding of the problem-solving capacity of computers in our world.
 	
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).
 	
8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
 	
NJSLS – 21st Century Life and Career Skills:
9.3.IT-PRG.6 – Program a computer application using the appropriate computer language.
 	
9.3.IT-PRG.10 – Design, create and maintain a database.
 	
9.3.ST.2 – Use technology to acquire, manipulate, analyze and report data.
 	
9.3.ST-ET.4 – Apply the elements of the design process.
 	
9.3.ST-SM.1 – Apply science and mathematics to provide results, answers and algorithms for engineering and technological activities.
	All types of variables in Python are objects that can be either mutable or immutable.
	
	Other array-like structures exist, including immutable tuples and mutable dictionaries.
	
	Python allows programmers to dynamically allocate arrays at runtime.
	
	In Python a string behaves as an array of characters and can be processed using standard array processing techniques or using standard string library functions.
		Write simple programs that verify the mutability or immutability of common object types, including:
		• ints
		• floats
		• boolean
		• arrays
		• tuples
		• dictionaries
		• strings
		• dictionaries
		
		Use tuples similarly to arrays, but when immutability or unchanging behavior is required.
		
		Use dictionaries to generate key value pairs.
		
		Characterize dictionary similarities and differences to arrays.
		
		Write programming code to dynamically allocate values to an array at runtime.
		
		Write programming code to perform functions on strings, including:
		• Declaring and initializing strings
		• String input and output using input() and print()
		• Implementing standard string methods, including:
		  ○ len('string')
		  ○ s.lower() and s.islower()
		  ○ s.upper() and s.isupper()
		  ○ s.strip() and s.split('delim')
		  ○ s.find('string')
		  ○ s.count('string')
		  ○ s.join(list)

	Vocabulary: array, element, index, subscript, memory address, reference, string, concatenate	

ASSESSMENT EVIDENCE: Students will show their learning by:

- Storing same-types data in an array structure rather than several distinct variables.
- Indicating instances where passed data retains a value after processing versus when that value is changed.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students use user input to move a text character through a "maze" defined by cells in a two-dimensional array.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT IV: Python Data Structures and Strings

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
4 Weeks	Unit 4: Python Data Structures and Strings 1) Arrays a. One-dimensional arrays b. Array initialization c. Arrays as arguments d. Two-dimensional arrays 2) Mutable and Immutable Types 3) Array-like structures a. Tuples and immutability b. Dictionaries and mutability 4) Character Strings a. String initialization b. String input and output c. String processing	Python and Integrated Development Environment: https://www.python.org/getit/ Python 3.6.6rc1 Documentation: https://docs.python.org/3/index.html Online Python Coding Practice: http://codingbat.com/python

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT V: Python Classes and Introduction to Object-Oriented Programming

<table>
<tr><td rowspan="6">STANDARDS / GOALS:

CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 – Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 – Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17 – Systematically test and refine programs using a range of test cases.

3B-AP-09 – Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.

3B-AP-14 – Construct solutions to problems using student-created components, such as procedures, modules and/or objects.</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Object-oriented programming languages allow programmers to define customized data types.</td><td>• How can we create custom data types to solve problems?</td></tr>
<tr><td>Advanced computer programs involve the interaction of several classes with one another, including their inherent attributes and behaviors.</td><td>• How do an object’s attributes and behaviors translate into programming language?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

A Python class acts as a blueprint for a real-world object, and thereby defines a new data type.

Instantiation of an object in Python may be customized, meaning the values of several variables may be assigned different values for different objects.</td><td>Students will be able to:

Describe the attributes and behaviors of a real-world object, like a dog or a high school student.

Select appropriate existing data types for attributes of real-world objects.

Define Python classes, taking into account:
• How the class header is written
• Proper indentation
• Positioning within a Python script

Define __init__() methods for classes to customize object instantiation.

Use the ‘self’ keyword to refer to selected instances of created objects.</td></tr>
</table>

NJSLS – Technology: 8.2.12.E.1 – Demonstrate an understanding of the problem-solving capacity of computers in our world. 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). 8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements). **NJSLS – 21st Century Life and Career Skills:** 9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.ST.2 – Use technology to acquire, manipulate, analyze and report data.	Variables may be customized so that some attributes are shared among different instances of a class, and some are unique to different instances. **Vocabulary**: class, object, instance, instantiation, dot notation, self, instance variable, class variable, method	Distinguish between class and instance variables, including: • How each is instantiated • When each should be used Use dot notation to refer to the variables and methods of an object.

ASSESSMENT EVIDENCE: Students will show their learning by:

- Creating a Python class to represent a real-world object of their choosing.
- Using a self-generated Python class as a data type within a separate program.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students formalize the attributes and behaviors of a Pet and translate those to data types and methods for a class.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT V: Python Classes and Introduction to Object-Oriented Programming

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
2 Weeks	Unit 5: Python Classes and Introduction to Object-Oriented Programming 1) Class Attributes and Behaviors a. Attributes as variables b. Behaviors as methods 2) Class Definition a. __init__() method b. Variable locations 3) Variables a. Instance variables b. Class variables c. Dot notation	Python and Integrated Development Environment: https://www.python.org/getit/ Python 3.6.6rc1 Documentation: https://docs.python.org/3/index.html Online Python Coding Practice: http://codingbat.com/python

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT VI: Object-Oriented Programming in Java

<table>
<tr><td rowspan="8">STANDARDS / GOALS:

CSTA:
2-NI-05 – Explain how physical and digital security measures protect electronic information.

2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 – Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 – Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17 – Systematically test and refine programs using a range of test cases.

2-IC-23 – Describe tradeoffs between allowing information to be public and keeping information private and secure.</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Computer programmers serve as the link between human language and code that may be translated into machine language object code.</td><td>• How does a programmer use a computer program to get a computer to perform a task?</td></tr>
<tr><td>Object-oriented programming helps programmers write software solutions in Java by allowing definition of new data types.</td><td>• How can real-world objects be represented as Java classes?
• How do the definitions of classes and objects help the programmer avoid duplication of programming efforts?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Proper terminology is important for communicating a program’s functionality.</td><td>Students will be able to:

Describe and explain the function of all the major hardware computer components and their relationship in enabling the execution of software by the computer.

Explain how computers execute programs in machine language (binary code).

Explain the operation of compilers and interpreters in translating source code written in high-level languages like Java to machine language object code (bytecode).</td></tr>
<tr><td>A brief history of the Java language.</td><td>Describe the history of the Java programming language including:</td></tr>
</table>

3B-AP-09 – Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem. 3B-AP-14 – Construct solutions to problems using student-created components, such as procedures, modules and/or objects. 3B-AP-24 – Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems. **NJSLS – Technology:** 8.1.12.D.5 – Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs. 8.1.12.F.1 – Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and/or social needs. 8.2.12.B.4 – Investigate a technology used in a given period of history, e.g., the stone age, industrial revolution or information age, and identify their impact and how they may have changed to meet human needs and wants. 8.2.12.C.3 – Analyze a product or system for factors such as safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, and human factors engineering (ergonomics).	The benefits of using the Java programming language. The general principles of object-oriented programming, including encapsulation, inheritance, and polymorphism. Use an Integrated Development Environment (IDE) to edit, compile, and run Java programs.	• Java was originally developed by Sun Microsystems to run interactive television, but these electronics were too advanced for the digital cable industry at the time. • Language development was redirected to work with the Internet to allow people to download programs instead of just text and graphics. Explain the benefits of the Java programming language: • Platform independence. • Very popular language for software development. • Object-oriented and contains an extensive set of library classes. • It is robust. • It is a distributed language. • It is secure. Write classes that are: • Templates for defining characteristics and behaviors for all objects of a certain type. • Part of a class hierarchy. Create simple classes in an IDE. Write, debug, compile, and run simple Java programs including: • Simple output operations (PrintStream). • Qualified names. • Java Applets. • Java packages and libraries.

8.2.12.E.1 – Demonstrate an understanding of the problem-solving capacity of computers in our world. 8.2.12.E.2 – Analyze the relationships between internal and external computer components. 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). 8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements). **NJSLS – 21st Century Life and Career Skills:** 9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.ST-SM.3 – Analyze the impact that science and mathematics has on society.	Designing good programs in Java requires planning. **Vocabulary**: class, class inheritance, class hierarchy, compiler, interpreter, method, object, subclass, superclass, Unified Modeling Language (UML), semantic error, syntax error, instance of, import, packages, Java Virtual Machine (JVM)	Create objects that encapsulate certain variables (characteristics, values, attributes) and methods (behaviors, actions). Describe and explain whether given programming errors are logic (semantic) errors or errors are caught by the compiler (syntax errors). Plan simple programs on paper using Unified Modeling Language (UML) diagrams prior to beginning to code. Write simple programs utilizing Java style conventions and appropriate comments. Select identifiers, variables, methods, and class names using naming conventions and avoiding reserved words (keywords).

ASSESSMENT EVIDENCE: Students will show their learning by:

- Leading a class presentation on the differences between the Java programming language and the Python language with which they have become familiar.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students will look for opportunities to extend their Pet objects from the previous unit to more-specific animals.
- Students will complete an assignment they have already done in Python in the new Java language.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT VI: Object-Oriented Programming in Java

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
3 Weeks	Unit 6: Object-Oriented Programming in Java 1) Computers, Objects, and Java a. Computers, networks, internet b. Programming languages c. Why Java 2) Java Programs: Design and Development a. Designing good programs b. Edit, compile, run programs c. Program output	Integrated Development Environment: http://www.drjava.org Integrated Development Environment: http://www.eclipse.org Integrated Development Environment: http://www.BlueJ.org Integrated Development Environment: https://www.jetbrains.com/idea/ Java Platform, Standard Edition 7, API Specification: https://docs.oracle.com/javase/7/docs/api/ Online Java Coding Practice: http://codingbat.com/java An Object-Oriented Bedtime Story, Joseph Bergin, Pace University, http://csis.pace.edu/~bergin/Java/OOStory.html Morelli, Ralph and R. Walde. *Object-Oriented Problem Solving, Java, Java, Java*. Third Edition, Hartford, Conn.: Trinity College, 2012. Open source online version: http://www.cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT VII: Java Basics and Selection Control Structures

<table>
<tr><td rowspan="8">STANDARDS / GOALS:

CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 – Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 – Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17 – Systematically test and refine programs using a range of test cases.

2-AP-19 – Document programs in order to make them easier to follow, test, and debug.

3A-AP-17 – Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Defining classes and declaring objects are critical in object-oriented programming for defining and instantiating new data types.</td><td>• How does effective class design help avoid code duplication?</td></tr>
<tr><td>Selection structures allow a program to make decisions while the program is executing.</td><td>• Why is it important for a program to be able to make run-time decisions?</td></tr>
<tr><td>Different user interfaces, such as text-based and Graphical (GUIs), can improve the user experience depending on the application.</td><td>• What are the advantages of using Graphical User Interfaces in comparison to using simple input/output?</td></tr>
<tr><td>The order of precedence for operators determines how mathematical expressions are evaluated.</td><td>• Why is an understanding of operator precedence essential for problem solving?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Classes are blueprints for objects.</td><td>Students will be able to:

Create classes and write programs that:
• Utilize String objects and methods from the String class.
• Illustrate the relationship between classes and objects.
• Utilize static and instance elements of a class
• Utilize java.util.Scanner for user input.
• Utilize methods that return a value.
• Utilize instances of objects that require qualified names (dot operator).</td></tr>
</table>

3A-AP-21 – Evaluate and refine computational artifacts to make them more usable and accessible. 3B-AP-16 – Demonstrate code reuse by creating programming solutions using libraries and APIs. 3B-AP-21 – Develop and use a series of test cases to verify that a program performs according to its design specifications. **NJSLS – Technology:** 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). 8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements). **NJSLS – 21st Century Life and Career Skills:** 9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.IT-PRG.7 – Demonstrate software testing procedures to ensure quality products. 9.3.IT-PRG.10 – Design, create and maintain a database. 9.3.ST.2 – Use technology to acquire, manipulate, analyze and report data. 9.3.ST-SM.1 – Apply science and mathematics to provide results, answers	Primitive types and objects are both means for data storage. Information may be passed to methods either by value or by reference. Correct implementation of selection control structures in java. Classes and methods are used in Java to accept user input, provide output, and create effective user interfaces.	Write, compile, and run an applet that utilizes the Graphics object. Write classes and methods that utilize objects as well as primitive datatypes. Write programs that: • Use parameters and arguments to pass data to an object. • Retrieve information from an object using a return value. • Use constructor methods to instantiate objects. • Pass by value for primitive data types. • Pass by reference for objects. • Take advantage of polymorphism and inheritance to minimize duplicate code. • Utilize overloaded constructors and methods. Trace existing code to determine changes to stored data based upon whether parameters are passed by value or by reference. Write programs and methods that will properly utilize selection control structures including: • If statements. • Else-if statements. • Nested if statements. • While statements. • Switch statements. Write programs and methods requiring user input and program output utilizing:

and algorithms for engineering and technological activities.		• Command line interface. • Graphical User Interface (GUI) using Swing components. • Java Applets.
	Boolean, numeric, and character data types and their associated operators in Java programs.	Write programs and methods that utilize various Java data types and operators. Programs should demonstrate: • Proper use of primitive data types. • Use of class constants and class methods. • Conversions from one data type to another using the cast operator and wrapper classes. • Use of methods from the Math class and the NumberFormat class. • Effective use of information hiding principles.
	Proper implementation of operator precedence, data types, and associativity to solve mathematical problems using Java programs.	Write methods utilizing complex mathematical algorithms, formulas, and relational, logical, and mathematical operators in the code. Trace existing code segments and use written assignments to determine the result of complex mathematical problems, algorithms, and formulas as they are translated to Java code.
	Vocabulary: local variable, method call, return, pointer, reference variable, accessor method, mutator method, class scope, pedigree, method overloading, selection statement, multiway selection, override, polymorphism, formal parameter, abstract class, command-line interface, container, inheritance, action listener, Swing class, wrapper class, unary operator, binary operator, precedence order, associativity, cast operation, named constant, operator overloading, short-circuit evaluation, Unicode	

ASSESSMENT EVIDENCE: Students will show their learning by:

- Formalizing the primitive data types of numeric and character computer data that previously did not require formalization.
- Tracing existing code to describe output and program execution.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students write a program to accept transactions from a user and give simple financial information.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT VII: Java Basics and Selection Control Structures

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
6 Weeks	Unit 7: Java Basics and Selection Control Structures 1) Objects: Using, Creating, Defining a. Using string objects b. Class definition c. Objects and primitive data types d. Static and instance elements of a class e. Design a simple class in Java f. Principles of object-oriented programming 2) Methods: Communicating with Objects a. Passing information to an object b. Constructors and retrieving information from an object c. Pass by value and pass by reference d. Introduction to inheritance and polymorphism 3) Input/Output: Designing the User Interface a. User interface b. Command-line interface c. Graphical User Interface (GUI) d. Event-driven programming e. Java applets 4) Java Data and Operators a. The role of data in effective program design b. Primitive data types c. Information hiding d. Class constants and class methods e. Java Math and NumberFormat classes f. Data conversions	Integrated Development Environment: http://www.drjava.org Integrated Development Environment: http://www.eclipse.org Integrated Development Environment: http://www.BlueJ.org Integrated Development Environment: https://www.jetbrains.com/idea/ Java Platform, Standard Edition 7, API Specification: https://docs.oracle.com/javase/7/docs/api/ Online Java Coding Practice: http://codingbat.com/java An Object-Oriented Bedtime Story, Joseph Bergin, Pace University, http://csis.pace.edu/~bergin/Java/OOStory.html Morelli, Ralph and R. Walde. *Object-Oriented Problem Solving, Java, Java, Java*. Third Edition, Hartford, Conn.: Trinity College, 2012. Open source online version: http://www.cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT VIII: Repetition Control Structures, Strings, and Arrays

<table>
<tr><td rowspan="8">STANDARDS / GOALS:

CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11 – Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17 – Systematically test and refine programs using a range of test cases.

3A-DA-10 – Evaluate the tradeoffs in how data elements are organized and where data is stored.

3A-AP-14 – Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-15 – Justify the selection of specific control structures when tradeoffs involve implementation, readability, and</td><td>ENDURING UNDERSTANDINGS</td><td>ESSENTIAL QUESTIONS</td></tr>
<tr><td>Programmers use repetition structures to control the execution of a program’s repeated actions.</td><td>• What determines the order in which a computer program is executed?</td></tr>
<tr><td>Strings are computer programming objects used for text input, manipulation, and output that allow a program to give a sensible response in the user’s language.</td><td>• How can we manage and manipulate text and characters in a computer program?</td></tr>
<tr><td>Arrays are storage structures in programming languages that help manage large quantities of data.</td><td>• How can arrays and other objects be used to manage large amounts of data?</td></tr>
<tr><td>KNOWLEDGE</td><td>SKILLS</td></tr>
<tr><td>Students will know:

Problem solutions may require repetition and various loop designs.

The principles of effective loop design.</td><td>Students will be able to:

Write methods and code that implement:
• Counting loops with the for statement.
• Conditional loops with the:
o while statements.
o do-while statements.

Write pseudocode to plan how loops will be used in advance of coding.

Write code using nested loops to output given patterns.

Trace existing code segments to determine output of various counting and conditional loops.</td></tr>
</table>

program performance, and explain the benefits and drawbacks of choices made. 3A-AP-21 – Evaluate and refine computational artifacts to make them more usable and accessible. 3B-AP-12 – Compare and contrast fundamental data structures and their uses. 3B-AP-13 – Illustrate the flow of execution of a recursive algorithm. **NJSLS – Technology:** 8.2.12.E.1 – Demonstrate an understanding of the problem-solving capacity of computers in our world. 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). 8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements). **NJSLS – 21st Century Life and Career Skills:** 9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.IT-PRG.10 – Design, create and maintain a database. 9.3.ST.2 – Use technology to acquire, manipulate, analyze and report data. 9.3.ST-ET.4 – Apply the elements of the design process.	Strings can be manipulated to solve problems in Java. String processing algorithms may require looping behavior. Array structures are used to solve problems that require a collection of data.	Trace code to determine if it represents an infinite loop and determine how to modify code to obtain the desired output. Create classes and write programs that utilize string objects and methods from the string class. Write programs that utilize methods from the string class to manipulate strings including: • locating characters or text within a string. • concatenating strings. • retrieving parts of strings. • processing names and passwords. • accepting input using the Scanner and StringBuffer classes. • comparing strings. Write programs that utilize methods from the String class and loop design to: • do keyword and character searches. • process each character in a string. • count the number of a certain character within a string. Write code segments and programs that declare and initialize one-dimensional and two-dimensional arrays. Write code which requires the processing of elements in an array based on the subscript or index of each element including: • solving for the average of all elements in an array. • determining if two words are anagrams.

9.3.ST-SM.1 – Apply science and mathematics to provide results, answers and algorithms for engineering and technological activities.	**Vocabulary**: conditional loop, counting loop, do-while statement, infinite loop, loop entry condition, post-condition, precondition, sentinel bound, concatenation, delimited string, lexicographic order, string index, string literal, zero indexed, array, element, element type, data structure	• determining if a word or phrase is a palindrome . • using nested loops to traverse and process elements of 2D arrays.

ASSESSMENT EVIDENCE: Students will show their learning by:

- Executing simple searching and sorting algorithms with strings.
- Completing CodingBat practice assignments for quick string processing.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students create a program to accept a list of names and perform degrees of alphabetization.
- Students create a fractal using Java Graphics and recursion.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT VIII: Repetition Control Structures, Strings, and Arrays

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
7 Weeks	Unit 8: Repetition Control Structures, Strings, and Arrays 1) Repetition Control Structures a. Repetition control structures b. Counting loops c. Conditional loops 2) Strings and String Processing a. String basics b. String methods c. Java.lang.StringBuffer d. Processing each character in a string e. Comparing strings 3) Arrays and Array Processing a. One-dimensional arrays b. Array processing c. Two-dimensional arrays	Integrated Development Environment: http://www.drjava.org Integrated Development Environment: http://www.eclipse.org Integrated Development Environment: http://www.BlueJ.org Integrated Development Environment: https://www.jetbrains.com/idea/ Java Platform, Standard Edition 7, API Specification: https://docs.oracle.com/javase/7/docs/api/ Online Java Coding Practice: http://codingbat.com/java An Object-Oriented Bedtime Story, Joseph Bergin, Pace University, http://csis.pace.edu/~bergin/Java/OOStory.html Morelli, Ralph and R. Walde. *Object-Oriented Problem Solving, Java, Java, Java*. Third Edition, Hartford, Conn.: Trinity College, 2012. Open source online version: http://www.cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT IX: Java Applications (Finch Robot and Greenfoot)

<table>
<tr>
<td rowspan="6">STANDARDS / GOALS:

CSTA:
2-AP-10 – Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-16 – Incorporate existing code, media, and libraries into original programs, and give attribution.

3A-AP-17 – Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3B-AP-09 – Implement an artificial intelligence algorithm to play a game against a human opponent or solve a problem.

NJSLS – Technology:
8.2.12.E.1 – Demonstrate an understanding of the problem-solving capacity of computers in our world.

8.2.12.E.2 – Analyze the relationships between internal and external computer components.

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).

8.2.12.E.4 – Use appropriate terms in conversation (e.g., troubleshooting,</td>
<td>ENDURING UNDERSTANDINGS</td>
<td>ESSENTIAL QUESTIONS</td>
</tr>
<tr>
<td>A robot’s behavior can be determined by a programmer writing Java code.</td>
<td>• How can a programmer make a robot interact with its environment?</td>
</tr>
<tr>
<td>Games and simulations can be designed by a programmer writing Java code.</td>
<td>• How can a programmer design a game that responds to player input?</td>
</tr>
<tr>
<td>KNOWLEDGE</td>
<td>SKILLS</td>
</tr>
<tr>
<td>Students will know:

The Java programming language may be applied to make the Finch robot perform various tasks.

The Java programming language may be used to design games, simulations, and graphical environments.

Vocabulary: Finch robot, obstacle detection system, light sensors, accelerometer, temperature sensor, gear motors, LED, buzzer</td>
<td>Students will be able to:

Write programming code to receive input from the obstacle detection system, light sensors, accelerometer, and/or temperature sensor to perform a task in response to the input.

Write programming code to generate output using the LEDs, buzzer, gear motors, and/or computer speakers based on code and specified input.

Write programming code to create a game with objects that interact with each other and their environment.</td>
</tr>
</table>

peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements). **NJSLS – 21st Century Life and Career Skills:** 9.2.12.C.3 – Identify transferable career skills and design alternate career paths. 9.3.IT-PRG.6 – Program a computer application using the appropriate computer language. 9.3.ST.2 – Use technology to acquire, manipulate, analyze and report data. 9.3.ST-ET.4 – Apply the elements of the design process. 9.3.ST-ET.5 – Apply the knowledge learned in STEM to solve problems. 9.3.ST-SM.1 – Apply science and mathematics to provide results, answers and algorithms for engineering and technological activities.		

ASSESSMENT EVIDENCE: Students will show their learning by:

- Generating tangible results of their coding through use of a real-world robot.
- Implement mechanics of a simple game using Greenfoot https://www.greenfoot.org/door.

KEY LEARNING EVENTS AND INSTRUCTION:

- Students write programming code to move a real-world robot around obstacles to a goal.
- Students use player, goal, and obstacle-type objects in a created game that can interact with one another.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java Honors
UNIT IX: Java Applications (Finch Robot and Greenfoot)

SUGGESTED TIME ALLOTMENT	CONTENT-UNIT OF STUDY	SUPPLEMENTAL UNIT RESOURCES
2 Weeks	Unit 9: Java Applications (Finch Robot and Greenfoot) 1) Greenfoot a. Objects b. Movement and key control c. Actors and methods d. Making and playing sound e. Adding an enemy f. Interacting objects g. Design game for others to play 2) Finch Robots a. Hardware/sensors b. Introduction c. Variables d. Conditionals e. Iteration f. Arrays g. Objects, methods, classes h. Program to perform a task	Greenfoot Resources: https://www.greenfoot.org/door Finch Robot Resources: http://finchrobot.com/ Integrated Development Environment: http://www.drjava.org Integrated Development Environment: http://www.eclipse.org Integrated Development Environment: http://www.BlueJ.org Integrated Development Environment: https://www.jetbrains.com/idea/ Java Platform, Standard Edition 7, API Specification: https://docs.oracle.com/javase/7/docs/api/ Morelli, Ralph and R. Walde. *Object-Oriented Problem Solving, Java, Java, Java*. Third Edition, Hartford, Conn.: Trinity College, 2012. Open source online version: http://www.cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java

APPENDIX A

RESOURCES:

Primary Textbooks:

Morelli, Ralph and R. Walde. *Object-Oriented Problem Solving, Java, Java, Java.* Third Edition, Hartford, Conn.: Trinity College, 2012.
Open source online version: http://www.cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf

Additional Resources:

Beiter, Mike, Brian Coltin, Somchaya Liemhetcharat. *An Introduction to Robotics with NAO*, 1st Edition, Aldebaran Robotics, 2013.

Choreographe Software used to Program NAO Robot: http://www.aldebaran.com/en/robotics-solutions/robot-software

Online Python Coding Practice:
PythonBat: http://codingbat.com/python

Java Integrated Development Environments:
DrJava: http://www.drjava.org
Eclipse: http://www.eclipse.org
BlueJ: http://www.BlueJ.org
IntelliJ: https://www.jetbrains.com/idea/

Java Platform, Standard Edition 7, API Specification: https://docs.oracle.com/javase/7/docs/api/

Online Java Coding Practice:
JavaBat: http://codingbat.com/java
Practice-It: https://practiceit.cs.washington.edu/

An Object-Oriented Bedtime Story, Joseph Bergin, Pace University, http://csis.pace.edu/~bergin/Java/OOStory.html

Greenfoot Resources: https://www.greenfoot.org/door

Finch Robot Resources: http://www.finchrobot.com/

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java

APPENDIX B

ASSESSMENT:

- Pre-assessment
- Quizzes
- Tests
- Programming assignments
- Program sets
- Homework
- Individual projects

APPENDIX C

Opportunities exist for interdisciplinary units with courses such as Physics, Business, Technology, Robotics, or other science and mathematics courses and electives.

RANDOLPH TOWNSHIP SCHOOL DISTRICT
Programming with Python and Java

APPENDIX D

Programming with Python and Java is the second of the required courses in the Computer Science and Programming Pathway, following AP Computer Science Principles and preceding AP Computer Science A. The complete sequence of courses, with both required and choice selections, is both below and available at https://www.rtnj.org/domain/1185.

Course	Credits	Recommended Year
AP Computer Science Principles (required)	5	9th or 10th
Programming with Python & Java Honors (required)	5	10th or 11th
AP Computer Science A (required)	5	10th, 11th, or 12th
Robotics (choice)	5	12th
Data Structures Honors (choice)	5	12th
Total Credits Needed for Pathways Distinction: 20		