Standard: Technology Literacy (2009)

8.1: Education Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaboratively and to create and communicate knowledge.

8.2: Technology Education, Engineering, and Design: All students will develop an understanding of the nature and impact of technology, engineering, technology design, and the designed world, as they relate to the individual, global, and the environment.

9.1: 21st Century Life and Career Skills: All students will demonstrate the creativity, critical thinking, collaboration, and problem-solving skills needed to function successfully as both global citizens and workers in diverse ethnic and organizational cultures.

Strand:

8.1.A: Technology Operations and Concepts
8.1.F: Critical Thinking, Problem Solving, and Decision Making

8.2.B: Design: Critical Thinking, Problem Solving, and Decision Making
8.2.F: Resources for a Technological World
8.2.G: The Designed World

9.1.A: Critical Thinking and Problem Solving
9.2.E: Communication and Media Fluency
9.2.F: Accountability, Productivity, and Ethics

Curriculum aligned with: 2009 New Jersey Core Curriculum Content Standards for 21st Century Skills (9.1 A-F)

21st Century Theme: *Global Awareness ☐,Financial, economic, business and entrepreneurial literacy ☐ Civic literacy ☐,Health literacy ☐Environmental Literacy X*

21st Century Skills: *Critical Thinking & Problem Solving ☐,Creativity and Innovation ☐,Collaboration, Teamwork and Leadership ☐,Cross-Cultural Understanding and Interpersonal Communications X Communication and Media Fluency ☐, Accountability, Productivity and Ethics ☐*

Interdisciplinary Connection: Math=MA, English=ELA, Science=SCI, Social Studies=SS, Physical Education=PE, Art=ART, Music=MU, Technology=TECH, World Language=WL Business = BU

Essential Questions	Enduring Understandings	Activities, Investigation, and Student Experiences
• Why do we need computer language? • What is an algorithm? • What does "efficiency" mean when programming? • Why constrain to 4 commands? • What commands would you have liked to have available for the 3x3 square challenge? • What are APIs? • What are Random numbers? • What are events? • What are APPs?	*Students will understand….* • The use of computer language. • When writing code to program a computer, you are always limited by the constraints of the programming language you choose to use. • students learn to define and call procedures • Students use a worksheet to learn about top-down design, and then on paper, design a solution to a new turtle drawing challenge with a partner. Students will then move to Code Studio to implement their solution. • The process of creating software begins long	• Lab 1: Students construct a simple arrangement of 5 or so LEGO blocks. Students create written instructions for how to build their arrangement from the blocks. Students trade instructions and attempt to follow them. • Lab 2: Introduction to App Lab Turtle programming problems in Code Studio (up to 3x3 grid) • Lab 3: Use App Lab's APIs to complete a set of drawing challenges. • Lab 4: Digital Assistant Project In this lesson, students complete the digital assistant project in which they synthesize their abilities to write string functions and complex conditionals. • Lab 5: Students will write with draw. • Lab 6: Mini Project - Students work in groups of 3 or 4 and begin by identifying a scene they wish to create. They then use Top-Down Design to identify the high-level functions

● What are program flow and logic? ● What are loops? ● What are arrays? ● What are functions? ● What are return values? ● What is a Canvas?	before the first lines of code are written. ● to read App Lab's API documentation and will use functions that accept parameters in order to complete a series of drawing puzzles which require them to make use of the AppLab API documentation to learn new drawing commands. ● An API is a reference guide which catalogs and explains the functionality of a programming language. ● use random values and looping to create variation in their drawings and quickly duplicate objects they wish to appear in their digital scenes many times. ● Collaborate to break down a complex programming problem	necessary to create that image, and assign these components to individual members. After programming their individual portion, students combine all of their code to compose the whole scene. The project concludes with reflection questions similar to those students will see on the AP® Performance Tasks. **Spot Light On:** *Acknowledge every student's comment or response, even if it's incorrect.* **Unit Project:** SS ● Lab 7: students will design an app based off of one they have previously worked on in the programming unit. **Modifications and/or Accommodations:** ● **Special Education:** Utilize a multi-sensory (VAKT) approach during instruction, provide alternate presentations of skills by varying the method (repetition, simple explanations, additional examples, modeling, etc.), modify test content and/or format, allow students to retake test for additional credit, provide additional times and preferential seating as needed, review, restate and repeat directions, provide study guides, and/or break assignments into segments of shorter tasks. ● **English Language Learners:** Extend time requirements, preferred seating, positive reinforcement, check often for understanding/review, oral/visual directions/prompts when necessary, supplemental materials including use of online bilingual dictionary, and modified assessment and/or rubric. ● **Students at Risk of School Failure:** Deliver instruction utilizing varied learning styles including audio, visual, and tactile/kinesthetic, provide individual instruction as needed, modify assessments and/or rubrics, repeat instructions as needed. **Gifted Students:** Create an enhanced set of introductory activities, integrate active teaching/learning opportunities, incorporate authentic components, propose

	into its component parts. ● Write programs that address one component of a larger programming problem and integrate with other similarly designed programs. ● Use code written by other programmers to complete a larger programming task. ● Abstraction is an important tool in programming, not only because it allows individual programmers to break down complex problems, but because it enables effective forms of collaboration.	interest-based extension activities, and connect student to related talent development opportunities.
Content Statements	**Cumulative Progress Indicators**	

Students will know… ● Why is is important to have computer language. ● Use App Lab to write programs that create simple drawings with "turtle graphics." ● Solve simple programming challenges when the set of allowed commands is constrained. ● Work with a partner to program a turtle task that requires about 50 lines of code. ● Explain considerations that go into "efficiency" of a program. ● Justify or explain choices made when programming a solution to a turtle task. ● Use a "top-down" problem-solving	● Tests ● Quizzes ● Practice problems for homework ● Projects ● Worksheets ● In-class programs ● Labs	

approach to identify sub-tasks of a larger programming task. ● Write a complete program with functions that solve sub-tasks of a larger programming task. ● Explain how functions are an example of abstraction. ● Define an API as the set of commands made available by a programming language. ● Use parameters to provide different values as input to procedures when they are called in a program. ● Use API documentation to assist in writing programs. ● Identify appropriate situations in a		

program for using a loop. ● Use a loop in a program to simplify the expression of repeated tasks. ● Use random values within a loop to repeat code that behaves differently each time it is executed.		
Desired Results		
● Read about societal challenges of the Internet and relate to personal experiences. ● Create a set of instructions in human language for building a simple LEGO block arrangement. ● Assess the clarity of a set of instructions expressed in human language. ● Describe the ambiguities inherent in human language and the ways programming languages seek to remove those ambiguities. ● Identify connections between the ability to program and the ability to solve problems. ● Explain why and how functions can make code easier to read and maintain. ● Recognize functions in programs as a form of abstraction.		

• Define and call simple functions that solve turtle drawing tasks. • Write a program that solves a turtle drawing problem using multiple levels of abstraction (i.e. functions that call other functions within your code).	
Standards for Mathematical Practices	Teacher Resources
1. Make sense of problems and persevere in solving them. 2. Reason abstractly and quantitatively. 3. Construct viable arguments and critique the reasoning of others. 4. Model with mathematics. 5. Use appropriate tools strategically. 6. Attend to precision. 7. Look for and make use of structure. 8. Look for and express regularity in repeated reasoning.	https://learnzillion.com https://www.khanacademy.org/ https://www.desmos.com/

LGBT and Disabilities Law: *N.J.S.A. 18A:35-4.35*

Kate Hutton

The mission is to ensure that every student is able to see themselves in our rich and diverse history.

Social and Emotional Learning: ***Competencies***	**Social and Emotional Learning:** ***Sub-Competencies***
Self-Awareness Social Awareness Self-Management Relationship Skills Responsible Decision-Making	● Recognizing the importance of self-confidence in handling daily tasks and challenges. ● Demonstrate an awareness of the expectations for social interactions in a variety of ways. ● Demonstrate an understanding of the need for mutual respect when viewpoints differ. ● Recognize the skills needed to establish and achieve personal and educational goals. ● Utilize positive communication and social skills to interact effectively with others. ● Develop, implement, and model effective problem solving and critical thinking skills.

New Jersey Legislative Statutes and Administrative Code (place an "X" before each law/statute if/when present within the curriculum map)									
	Amistad Law: *N.J.S.A. 18A 52:16A-88*		Holocaust Law: *N.J.S.A. 18A:35-28*	**x**	LGBT and Disabilities Law: *N.J.S.A. 18A:35-4.35*	**x**	Diversity & Inclusion: *N.J.S.A. 18A:35-4.36a*		Standards in Action: *Climate Change*