

Unit 2: Computing and Society: Transitions to Text

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

UNIT RATIONALE

Unit 2 continues to reinforce coding fundamentals as students are gradually introduced to text-based programming. In this unit, students will explore the impacts of computer science on our society and bring coding off the screen and into the physical world. Students will learn how images can be used to make decisions in programs and explore real-world applications and innovations that will shape our future.

ESSENTIAL QUESTIONS

How realistic is it to expect coding professionals to be experts on all programming languages? What are they really experts at?

Why are collections (like arrays or lists) an essential concept in programs?

How has mobile, wireless, and networked computing had an impact on innovation throughout the world?

Why are diverse perspectives and user stories so important when developing an app?

What are some advantages and challenges of cloud computing?

STANDARDS

NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

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

8.1.12.CS.1:

Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.

8.1.12.CS.4:

Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

8.1.12.IC.1:

Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.

8.1.12.DA.2:

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

8.1.12.AP.1:

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

8.1.12.AP.4:

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

8.1.12.AP.5:

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

8.1.12.AP.6:

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

8.1.12.AP.8:

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

8.1.12.AP.9:

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

8.2.12.ED.5:

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

8.2.12.ITH.1:

Analyze a product to determine the impact that economic, political, social, and/or cultural factors have had on its design, including its design constraints.

8.2.12.ITH.3:

Analyze the impact that globalization, social media, and access to open source technologies has had on innovation and on a society's economy, politics, and culture.

8.2.12.NT.1:

Explain how different groups can contribute to the overall design of a product.

8.2.12.EC.3:

Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.

8.2.12.ETW.4:

Research historical tensions between environmental and economic considerations as driven by human needs and wants in the development of a technological product and present the competing viewpoints.

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.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.8	Evaluate and refine computational artifacts to make them more usable and accessible.
CS.9-12.8.1.12.AP.9	Collaboratively document and present design decisions in the development of complex programs.
CS.9-12.8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.

CS.9-12.8.1.12.CS.4	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
CS.9-12.8.1.12.DA.2	Describe the trade-offs in how and where data is organized and stored.
CS.9-12.8.1.12.IC.1	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
CS.9-12.8.2.12.EC.3	Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.
CS.9-12.8.2.12.ED.5	Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
CS.9-12.8.2.12.NT.1	Explain how different groups can contribute to the overall design of a product.
CS.9-12.8.2.12.ETW.4	Research historical tensions between environmental and economic considerations as driven by human needs and wants in the development of a technological product and present the competing viewpoints.
CS.9-12.8.2.12.ITH.1	Analyze a product to determine the impact that economic, political, social, and/or cultural factors have had on its design, including its design constraints.
CS.9-12.8.2.12.ITH.3	Analyze the impact that globalization, social media, and access to open source technologies has had on innovation and on a society's economy, politics, and culture.

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

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

NEW JERSEY STUDENT LEARNING STANDARDS: COMPUTER SCIENCE AND DESIGN THINKING

See content area standards.

PRE-ASSESSMENTS

Loop review: Step 2 of Activity 2.1.1

INSTRUCTIONAL PLAN

MODULE 1

Lesson 2.1

Lesson 2.1 Transitions to Text Based Coding

Block-based programming is a great way to introduce coding fundamentals, but many students want to know, “What is happening inside those blocks?” Lesson 2.1 introduces students to the idea of a lower level of abstraction in a programming language. Students will develop in an environment that allows them to create in blocks, but see that same code in a text-based language.

Activity 2.1.1 Transitioning from Blocks to Text (2 days)

Activity 2.1.2 Dead Reckoning (3 days)

Activity 2.1.3 Coding Fundamentals: Lists (3 days)

Activity 2.1.4 Coding Fundamentals: 2D Lists (4 days)

Project 2.1.5 Map it, Drive it (5 days)

12.9.3.MN-QA.4	Employ project management processes using data and tools to deliver quality, value-added products.
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.8	Evaluate and refine computational artifacts to make them more usable and accessible.
CS.9-12.8.1.12.AP.9	Collaboratively document and present design decisions in the development of complex programs.
CS.9-12.8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
CS.9-12.8.1.12.CS.4	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
CS.9-12.8.1.12.DA.2	Describe the trade-offs in how and where data is organized and stored.
CS.9-12.8.1.12.IC.1	Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
CS.9-12.8.2.12.ED.5	Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).

Activity 2.1.1

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.1.1 Compare and contrast lower-level programming languages with higher-level programming languages Learn basic rules related to programming with syntax Get started with the interactive code editor Develop programs independently that uncover what is abstracted block-based programming languages
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Working algorithms in Trinket
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students are transitioning from block-based programming to text-based. They will write simple algorithms that: <ul style="list-style-type: none">• count down from 10 to 0• ask the user for their name• get and store input from the user• concatenate strings• use lists to store variables and find the length of the lists• import a math library so they can create a random number• create a guessing game• take a user-inputted number of seconds and convert it into minutes and seconds• calculate the tire circumference based on a user-inputted radiu
Suggested Modifications	English Language Learners Adjusted Speech: The teacher changes speech patterns to increase student comprehension. This could include facing the students, paraphrasing, clearly indicating the most important ideas and speaking more slowly. Visuals: The teacher uses graphics, pictures, visuals, and manipulatives. This helps ELL students better understand and comprehend the subjects at hand. Front-Loading Vocabulary: The teacher front loads vocabulary. This means providing students with a list of important vocabulary words they will need to know for a book, lesson, etc. prior to the lesson being taught. Including pictures to go with the vocabulary words is also very beneficial for the students. Students with Individualized Education Plans/504s Chunking: The teacher presents information in a way that makes easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded with excessive detail. The best way to deliver information is to organize it into meaningful units. Because students with special needs get overloaded easily, chunking is an effective strategy to use with them. Checking for Understanding: It is important to constantly check understanding, especially for students who have accommodations. Teachers want to make sure students understand the concepts being covered in a way that makes sense to them.

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

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

Gifted & Talented Strategies

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

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

Students at Risk of School Failure

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

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

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

Increase One to One Time: When other students are working, always touch base with your students at risk and find out if they're on track or needing some additional support. A few minutes here and there will go a long way to intervene as the need presents itself.

Contracts: It helps to have a working contract between you and your students at risk. This helps prioritize the tasks that need to be done and ensure completion happens. Each day write down what needs to be completed, as the tasks are done, provide a checkmark or happy face. The goal of using contracts is to eventually have the student come to you for completion sign-offs.

Hands On: As much as possible, think in concrete terms and provide hands-on tasks. This means a child doing math may require a calculator or counters. The child may need to tape record comprehension activities instead of writing them. A child may have to listen to a story being read instead of reading it him/herself.

Tests/Assessments: Tests can be done orally if needed. Break tests down in smaller increments by having a portion of the test in the morning, another portion after lunch and the final part the next day.

Seating: Seat students near a helping peer or with quick access to the teacher. Those with hearing or sight issues need to be close to the instruction which often means near the front.

Activity 2.1.2

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.1.2 Apply coding fundamentals to devices that move in the real world Get started with VEXcode® V5 Learn how to program a VEX® Self-Driving Vehicle.
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	The self-driving vehicle is able to navigate a well-defined environment with no sensors.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Student task: The mechanical engineers in your start-up have tasked your development team with testing the interface between the software used to program the SDVs (self-driving vehicles) and the motors. Salma thinks this is a good opportunity for you to start using the hardware. “I’ve drawn up a few different test paths and calculations for you to try out. Let’s make sure these vehicles behave the way we expect them to.”
Suggested Modifications	See Activity 2.1.1

Activity 2.1.3

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.1.3 Use lists in a program Modify and develop a program collaboratively that uses lists
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Self-driving vehicles are able to complete the spiral sequence successfully.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will work with their teams to determine a spiral sequence using nested loops to navigate from the center square on the grid outward in a spiral fashion.

Suggested Modifications	See Activity 2.1.1
--------------------------------	--------------------

Activity 2.1.4

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.1.4 Use two-dimensional lists in a program Collaboratively modify and develop a program that uses two-dimensional lists
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Students use a 2D list to map out parking lot aisles—where vehicle can drive, filled spots, and available parking spots for their SDV.
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will use two-dimensional (2D) lists to create a smiley face program that will display on the SDV screen. Then they will use a list to map physical locations on a grid to create a parking lot with some spaces already taken.
Suggested Modifications	See Activity 2.1.1

Project 2.1.5

Student Learning Intentions (SLI) WALT: (We are learning to...)	Project 2.1.5 Use lists in a program Modify and develop a program collaboratively that uses lists
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Students are able to navigate the three sample world layouts given
Formative Assessment (drives instructional decisions)	Activities and projects Screenshots of code Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will be shown 9 possible grids that they will need to program their SDV to be able to navigate through. They will be given 3 to test but they will not know in advance which 3 it will be. They therefore have to work in a team to create an algorithm that works for all possible situations.
Suggested Modifications	See Activity 2.1.1

Lesson 2.2

Lesson 2.2 Computing and Careers in Our Society

Just as clicks of a button or “swipes” of a screen are used to trigger events in an app, today, images are becoming increasingly important as a way to make decisions in programming. In this lesson, students will explore image processing and other innovations that are changing our society. Students will also begin to investigate the wide range of careers in computer science and how computational thinking is an important part of the majority of professions today and in the future.

Activity 2.2.1 Careers, Innovation, and Ethics in Computer Science (4 days)

Activity 2.2.2 Image Processing: Identification (4 days)

Activity 2.2.3 Decisions from Images (4 days)

Project 2.2.4 Image Processing: Navigation and Collision Avoidance (7 days)

12.9.3.MN-QA.4	Employ project management processes using data and tools to deliver quality, value-added products.
CS.9-12.8.2.12.EC.3	Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.
CS.9-12.8.2.12.ED.5	Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
CS.9-12.8.2.12.NT.1	Explain how different groups can contribute to the overall design of a product.
CS.9-12.8.2.12.ETW.4	Research historical tensions between environmental and economic considerations as driven by human needs and wants in the development of a technological product and present the competing viewpoints.
CS.9-12.8.2.12.ITH.1	Analyze a product to determine the impact that economic, political, social, and/or cultural factors have had on its design, including its design constraints.
CS.9-12.8.2.12.ITH.3	Analyze the impact that globalization, social media, and access to open source technologies has had on innovation and on a society’s economy, politics, and culture.
TECH.8.2.12.B.4	Investigate a technology used in a given period of history, e.g., stone age, industrial revolution or information age, and identify their impact and how they may have changed to meet human needs and wants.

Project 2.2.4

Student Learning Intentions (SLI) WALT: (We are learning to...)	Project 2.2.4 Decompose a project into smaller parts Apply coding fundamentals and iterative processes Develop a program as part of a team to solve a problem
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	The SDV avoids collisions while navigating a path.

Formative Assessment (drives instructional decisions)	Activities and projects Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Develop a program that uses image processing to identify marker and other self-driving vehicles to avoid collisions while navigating path.
Suggested Modifications	See Activity 2.1.1

Activity 2.2.3

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.2.3 Apply machine vision and conditionals Modify and develop a program collaboratively
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Successfully apply machine vision and conditionals to identify a predetermined color combination and provide the user with accurate feedback.
Formative Assessment (drives instructional decisions)	Activities and projects Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will create a program that takes vision sensor data, determines whether a user is showing the same color combination as what is stored, and provides the user instant feedback on the SDV LCD. Teams will each create a program that uses the vision sensor to capture and process color codes. Based on the signature that the sensor identifies, their program will display messages to the user on the SDV LCD.
Suggested Modifications	See Activity 2.1.1

Activity 2.2.2

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.2.2 Learn what information can be provided by machine vision system and object recognition software Apply machine vision to identify and differentiate between objects Modify and develop a program collaboratively to identify objects and patterns using machine vision
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Successful creation of unique identifiers to increase the likelihood the SDV vision sensor will detect and differentiate between objects

Formative Assessment (drives instructional decisions)	Activities and projects Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will learn how to record certain color signatures or pattern from a vision sensor. Then, using these specific color signatures and color codes, they will create algorithms that trigger events when those color patterns are detected.
Suggested Modifications	See Activity 2.1.1

Activity 2.2.1

Student Learning Intentions (SLI) WALT: (We are learning to...)	Activity 2.2.1 Develop a computational artifact independently to explore an innovation, career, or ethical consideration of computer science that interest you.
Student Learning Strategies	Journaling TEMP Charts (Term, Example, Meaning, Picture) Collaboration Cooperative Learning Pair Programming APB Approach (Activities, Projects, Problems) Class Discussions
Success Criteria	Presentation of the computational artifact that provides information about the computer science field or innovation they are interested in.
Formative Assessment (drives instructional decisions)	Activities and projects Vocabulary quizzes Call out responses Conclusion questions
Activities and Resources	Students will watch the Ted Talk: What moral decisions should driverless cars make? Then they will complete the MIT Moral Machine activity. They will discuss their results with the elbow partner. Next, students will identify an innovation and career they are interested in. They will do research and capture information about the career or innovation that they will creatively share with the class.
Suggested Modifications	See Activity 2.1.1

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

CCSS - English-Language Arts

Key Ideas and Details:

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

Integration of Knowledge and Ideas:

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

Production and Distribution of Writing:

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

Research to Build and Present Knowledge:

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

Range of Writing:

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

CCSS - Mathematics

Reason quantitatively and use units to solve problems:

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

Create equations that describe numbers or relationships:

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

Analyze functions using different representations:

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

Apply geometric concepts in modeling situations:

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

Calculate expected values and use them to solve problems:

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

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

English Language Arts

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

Social Studies

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

World Language

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

Math

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

occupying careers

Fine & Performing Arts

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

Science

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

REFLECTIONS
