Coding (CS Discoveries)

Unit Title: Unit 1: Problem Solving (Grades 7 & 8)

Stage 1: Desired Results

Standards & Indicators: **2020 NJSLS – Computer Science and Design Thinking**

8.2.8.ED.2: Identify the steps in the design process that could be used to solve a problem.

Career Readiness, Life Literacies and Key Skills

Standard	Performance Expectations	Core Ideas
9.4.8.CI.3	Examine challenges that may exist in the adoption of new ideas.	Gathering and evaluating knowledge and information from a variety of sources, including global perspectives, fosters creativity and innovative thinking.
9.4.8.CT.2	Develop multiple solutions to a problem and evaluate short- and long-term effects to determine the most plausible option.	Multiple solutions often exist to solve a problem.

Central Idea/Enduring Understanding:

Students will use a structured problem solving process to address problems and build a collaborative classroom environment where students view computer science as relevant, fun, and empowering.

Essential/Guiding Question:

What strategies and processes can I use to become a more effective problem solver?

Content:

Lesson 1- Intro to Problem Solving
Lesson 2- The Problem Solving Process
Lesson 3- Exploring Problem Solving

Skills(Objectives):

- Communicate and collaborate with classmates in order to solve a problem
- Identify different strategies used to solve a problem
- Iteratively improve a solution to a problem
- Given a problem, identify individual actions that would fall within each step of the problem solving process
- Identify useful strategies within each step of the problem solving process
- Apply the problem solving process to approach a variety of problems
- Assess how well-defined a problem is and use strategies to define the problem more precisely

Interdisciplinary Connections:

Science:

- MS-ETS1-1: Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit the possible solutions
- MS-ETS1-2: Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

Coding (CS Discoveries)

Stage 2: Assessment Evidence

Performance Task(s):

8.2.8.ED.2: Using the problem solving process
[Template] Copy of U1L03 - Activity G…

Other Evidence:

- Online assignments
- Exit Tickets
- Journal Entries
- Do Nows

Stage 3: Learning Plan

Learning Opportunities/Strategies:

Lesson 1- Intro to Problem Solving: Reflect on their experience with an activity and make connections to the types of problem solving they will be doing for the rest of the course.

Lesson 2- The Problem Solving Process: Relate the steps of the problem solving process to the problem from the previous lesson, then to a problem they are good at solving, then to a problem they want to improve at solving.

Lesson 3- Exploring Problem Solving: Apply the problem solving process to two different problems that are complex and poorly defined.

Resources:

- Lesson Presentations
- Google Docs
- Google Forms
- Google Classroom
- code.org

LGBT and Disabilities Law Resources:

- GLSEN Educator Resources
- Supporting LGBTQIA Youth Resource List
- Respect Ability: Fighting Stigmas, Advancing Opportunities

Differentiation

*Please note: Teachers who have students with 504 plans that require curricular accommodations are to refer to Struggling and/or Special Needs Section for differentiation

High-Achieving Students	On Grade Level Students	Struggling Students	Special Needs/ELL
Higher order thinking questions Differentiation of pacing and activities Differentiation of learning strategies: visual, auditory, kinetic and cooperative Enrichment and extension	Differentiation of learning strategies: visual, auditory, kinetic and cooperative Differentiating the lesson activities Lesson tutorials	Provide a highly structured, predictable learning environment Provide organizers Lessons designed to the style of learning that matches the student Cooperative Learning Positive reinforcement	Any student requiring further accommodations and/or modifications will have them individually listed in their 504 Plan or IEP. These might include, but are not limited to: breaking assignments into smaller tasks, giving directions through several channels (auditory, visual, kinesthetic, model), and/or small group instruction for reading/writing ELL supports should include, but

Coding (CS Discoveries)

		Lessons presentation available on google classroom Frequent check for understanding Break down task into manageable units One-on-one instruction Pair student with a high achieving student	are not limited to, the following:: Extended time Provide visual aids Repeated directions Differentiate based on proficiency Provide word banks Allow for translators, dictionaries

Coding (CS Discoveries)

Unit Title: Unit 2: Web Development (Grade 7)

Stage 1: Desired Results

Standards & Indicators: 2020 NJSLS – Computer Science and Design Thinking

8.1.8.AP.4: Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.
8.1.8.AP.5: Create procedures with parameters to organize code and make it easier to reuse.
8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.
8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.

Career Readiness, Life Literacies and Key Skills

Standard	Performance Expectations	Core Ideas
9.4.8.DC.1	Analyze the resource citations in online materials for proper use.	Detailed examples exist to illustrate crediting others when incorporating their digital artifacts in one's own work.
9.4.8.DC.2	Provide appropriate citation and attribution elements when creating media products.	Detailed examples exist to illustrate crediting others when incorporating their digital artifacts in one's own work.

Central Idea/Enduring Understanding:

Students should be able to create a digital artifact that uses multiple computer languages to control the structure and style of their content, and view computer science as a tool for personal expression. They should understand that different programming languages allow them to solve different problems, and that these solutions can be generalized across similar problems.

Essential/Guiding Question:

Why do people create websites?
How can text communicate content and structure on a web page?
How do I safely and appropriately make use of the content published on the internet?
What strategies can I use when coding to find and fix issues?

Content:

Lesson 1- Exploring Webpages
Lesson 2- Intro to HTML
Lesson 3- Headings
Lesson 4- Mini-Project: HTML Webpage
Lesson 5- Digital Footprint
Lesson 6- Styling Text with CSS
Lesson 7- Mini-Project: Your Personal Style
Lesson 8- Intellectual Property
Lesson 9- Using Images
Lesson 10- Websites for Expression
Lesson 11- Styling Elements with CSS
Lesson 12- Your Web Page- Prepare
Lesson 13- Project- Personal Web Page

Skills(Objectives):

- Identify the reasons someone might create a given website
- Identify the reasons someone might visit a given website
- Explain that HTML allows a programmer to communicate the way content should be structured on a web page
- Understand how to use lesson resources provided in Web Lab
- Write a simple HTML document that uses opening and closing tags to structure content
- Structure content into headings, subheadings, and paragraphs.

Coding (CS Discoveries)

	• Use a structured practice to collaboratively create a digital artifact. • Use heading tags to change the appearance of text on a web page. • Create a webpage using HTML • Understand and explain reasons that it is difficult to control who sees information published online. • Understand and justify guidelines for safely publishing information online. • Explain the differences between HTML and CSS in both use and syntax. • Link to an external style sheet. • Use CSS selectors to style HTML text elements. • Explain the purpose of copyright. • Identify the rights and restrictions granted by various Creative Commons licenses. • Add an image to a web page. • Follow copyright law, accurately attributing others when using their work. • Identify websites as a form of personal expression. • Create a CSS rule-set for the body element that impacts all elements on the page. • Use CSS properties to change the size, position, and borders of elements. • Create documentation that explains the design decisions of an artifact • Logically separate the content, structure and formatting of a digital artifact • Create a digital artifact

Interdisciplinary Connections:

English Language Arts:

- NJSLSA.W2: Write informative/explanatory texts to examine and convey complex ideas and information clearly and accurately through the effective selection, organization, and analysis of content.
- NJSLSA.W4: Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
- NJSLSA.W6: Use technology, including the internet, to produce and publish writing and to interact and collaborate with others.

Visual and Performing Arts:

- 1.5.8.Cr2a: Demonstrate persistence and willingness to experiment and take risks during the artistic process.
- 1.5.8.Cr2b: Demonstrate an awareness of ethical responsibility as applied to artmaking including environmental implications, responsibility in sharing images online, appropriation, and intellectual property ethics.
- 1.5.8.Cr2c: Apply, organize and strategize methods for design and redesign of objects, places, systems, images and words to clearly communicate information to a diverse audience.
- 1.5.8.Cr3a: Use criteria to examine, reflect on and plan revisions for a work of art, and create an artistic statement.

Stage 2: Assessment Evidence

Performance Task(s):	Other Evidence:

Coding (CS Discoveries)

Mini-Project: HTML Web Page https://studio.code.org/s/csd2-2021/lessons/4 Mini-Project: Personal Style - CSS Elements https://studio.code.org/s/csd2-2021/lessons/7 Project: Personal Webpage https://studio.code.org/s/csd2-2021/lessons/13	• Online assignments • Exit Tickets • Journal Entries • Do Nows • Reflection on assignments

Stage 3: Learning Plan

Learning Opportunities/Strategies:	**Resources:**
Lesson 1- Exploring Webpages: Explore a handful of sample web pages and describe how each of those pages is useful for users and how they might also serve their creators. **Lesson 2- Intro to HTML:** Introduce HTML as a solution to the problem of how to communicate both the content and structure of a website to a computer. **Lesson 3- Headings:** Practice using heading tags to create page and section titles and learn how the different heading elements are displayed by default. **Lesson 4- Mini-Project: HTML Webpage:** Design a web page, first identify the tags needed to implement them and then create the page in Web Lab. **Lesson 5- Digital Footprint:** Develop a set of guidelines to follow when putting information online. **Lesson 6- Styling Text with CSS:** Learn the basic syntax for CSS rule-sets and explore properties that impact HTML text elements. **Lesson 7- Mini-Project: Your Personal Style:** Design a web page, identify which CSS properties they will need, and create a web page in Web Lab. **Lesson 8- Intellectual Property:** Explore the various Creative Commons licenses as a solution to the difficulties of dealing with copyright. **Lesson 9- Using Images:** Learn how to add images to a web page using the <img> tag and how to cite the image source appropriately.	• Lesson Presentations • Google Docs • Google Forms • Google Classroom • Code.org • Web Lab - A browser-based tool for creating and publishing HTML and CSS web sites. LGBT and Disabilities Law Resources: • GLSEN Educator Resources • Supporting LGBTQIA Youth Resource List • Respect Ability: Fighting Stigmas, Advancing Opportunities

Coding (CS Discoveries)

Lesson 10- Websites for Expression: Create a resource for developing a personal website for the remainder of the unit. **Lesson 11- Styling Elements with CSS:** Learn additional CSS style properties focusing on non-text elements. **Lesson 12- Your Web Page- Prepare:** Engage in the "prepare" stage of the problem solving process to decide the elements and style of their web page. **Lesson 13- Project- Personal Web Page:** Create the webpage planned in Lesson 12.	

Differentiation

*Please note: Teachers who have students with 504 plans that require curricular accommodations are to refer to Struggling and/or Special Needs Section for differentiation

High-Achieving Students	On Grade Level Students	Struggling Students	Special Needs/ELL
Higher order thinking questions Differentiation of pacing and activities Differentiation of learning strategies: visual, auditory, kinetic and cooperative Enrichment and extension Code.org challenge levels	Differentiation of learning strategies: visual, auditory, kinetic and cooperative Differentiating the lesson activities Lesson tutorials	Provide a highly structured, predictable learning environment Provide organizers Lessons designed to the style of learning that matches the student Cooperative Learning Positive reinforcement Lessons presentation available on google classroom Frequent check for understanding Break down task into manageable units One-on-one instruction	Any student requiring further accommodations and/or modifications will have them individually listed in their 504 Plan or IEP. These might include, but are not limited to: breaking assignments into smaller tasks, giving directions through several channels (auditory, visual, kinesthetic, model), and/or small group instruction for reading/writing ELL supports should include, but are not limited to, the following:: Extended time Provide visual aids Repeated directions Differentiate based on proficiency Provide word banks Allow for translators, dictionaries

Coding (CS Discoveries)

		Pair student with a high achieving student Provide video tutorials	

Coding (CS Discoveries)

Unit Title: Unit 3: Interactive Animations (Grade 8)

Stage 1: Desired Results

Standards & Indicators: 2020 NJSLS – Computer Science and Design Thinking

8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values.
8.1.8AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
8.1.8.AP.4: Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.
8.1.8.AP.5: Create procedures with parameters to organize code and make it easier to reuse.
8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.
8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.

Career Readiness, Life Literacies and Key Skills

Standard	Performance Expectations	Core Ideas
9.4.8.CI.3	Examine challenges that may exist in the adoption of new ideas.	Gathering and evaluating knowledge and information from a variety of sources, including global perspectives, fosters creativity and innovative thinking.
9.4.8.CT.2	Develop multiple solutions to a problem and evaluate short- and long-term effects to determine the most plausible option.	Multiple solutions often exist to solve a problem.

Central Idea/Enduring Understanding:
Students should be able to create an interactive animation that includes basic programming concepts such as control structures, variables, user input, and randomness. Students should leave this unit viewing themselves as computer programmers, and see programming as a fun and creative form of expression.

Essential/Guiding Question:

What is a computer program?
What are the core features of most programming languages?
How does programming enable creativity and individual expression?
What practices and strategies will help me as I write programs?

Content:

Lesson 1- Programming for Entertainment
Lesson 2- Plotting Shapes
Lesson 3- Drawing in Game Lab
Lesson 4- Shapes and Parameters
Lesson 5- Variables
Lesson 6- Random Numbers
Lesson 7- Sprites
Lesson 8- Sprite Properties
Lesson 9- Text
Lesson 10- Mini-Project- Captioned Scene
Lesson 11- The Draw Loop
Lesson 12- Sprite Movement

Skills(Objectives):

- Identify how computer science is used in a field of entertainment
- Communicate how to draw an image in Game Lab, accounting for shape position, color, and order
- Reason about locations on the Game Lab coordinate grid
- Sequence code correctly to overlay shapes.
- Use a coordinate system to place elements on the screen.

Coding (CS Discoveries)

Lesson 13- Mini- Project- Animation Lesson 14- Conditionals Lesson 15- Keyboard Inputs Lesson 16- Mouse Input Lesson 17- Project- Interactive Card	● Use and reason about drawing commands with multiple parameters ● Identify a variable as a way to label and reference a value in a program ● Use variables in a program to store a piece of information that is used multiple times ● Generate and use random numbers in a program ● Update a value stored in a variable ● Create and use a sprite ● Use dot notation to update a sprite's properties ● Place text on the screen using a coordinate plane. ● Use arguments to control how text is displayed on a screen. ● Use a structured process to plan and develop a program. ● Explain how the draw loop allows for the creation of animations in Game Lab ● Use the draw loop in combination with the randomNumber() command, shapes, and sprites to make simple animations ● Identify which sprite properties need to be changed, and in what way, to achieve a specific movement ● Use the counter pattern to increment or decrement sprite properties ● Use conditionals to react to changes in variables and sprite properties ● Move sprites in response to keyboard input ● Use conditionals to react to keyboard input ● Respond to a variety of types of user input. ● Use an if-else statement to control the flow of a program. ● Apply an iterator pattern to variables or properties in a loop ● Sequence commands to draw in the proper order

Coding (CS Discoveries)

	• Use conditionals to react to keyboard input or changes in variables / properties

Interdisciplinary Connections:

Visual and Performing Arts:

- 1.5.8.Cr1a: Conceptualize early stages of the creative process, including applying methods to overcome creative blocks or take creative risks, and document the processes in traditional or new media.
- 1.5.8.Cr1b: Develop criteria, identify goals and collaboratively investigate an aspect of present-day life, using contemporary practice of art or design.
- 1.5.8.Cr2a: Demonstrate persistence and willingness to experiment and take risks during the artistic process.
- 1.5.8.Cr2b: Demonstrate an awareness of ethical responsibility as applied to artmaking including environmental implications, responsibility in sharing images online, appropriation, and intellectual property ethics.
- 1.5.8.Cr2c: Apply, organize and strategize methods for design and redesign of objects, places, systems, images and words to clearly communicate information to a diverse audience.
- 1.5.8.Cr3a: Use criteria to examine, reflect on and plan revisions for a work of art, and create an artistic statement.

Stage 2: Assessment Evidence

Performance Task(s):

Lesson 10: Mini-Project- Captioned Scene
https://studio.code.org/s/csd3-2021/lessons/10

Lesson 13: Mini-Project- Animation
https://studio.code.org/s/csd3-2021/lessons/13

Lesson 17: Project- Interactive Card
https://studio.code.org/s/csd3-2021/lessons/17

Other Evidence:

- Online assignments
- Exit Tickets
- Journal Entries
- Do Nows
- Reflection on assignments

Stage 3: Learning Plan

Learning Opportunities/Strategies:

Lesson 1- Programming for Entertainment: Explore how Computer Scienceand programming play a role in either a specific form of entertainment or as a vehicle of self expression.
Lesson 2- Plotting Shapes: Learn the difficulties of communicating how to draw with shapes and use tools in Game Lab.

Lesson 3- Drawing in Game Lab: Learn the basics of sequencing and debugging, and program images.

Lesson 4- Shapes and Parameters: Learn to draw with versions of ellipse() and rect() that

Resources:

- Lesson Presentations
- Google Docs
- Google Forms
- Google Classroom
- Code.org
- Game Lab - A browser-based JavaScript programming environment designed to create sprite-based drawings, animations and games, Enables students to switch between programming in blocks or text.

LGBT and Disabilities Law Resources:

- GLSEN Educator Resources
- Supporting LGBTQIA Youth Resource List

Coding (CS Discoveries)

include width and height parameters and to use the background() block. **Lesson 5- Variables:** Introduce variables as a way to label a number in a program or save a randomly generated value. **Lesson 6- Random Numbers:** Learn the randomNumber() block and how it can be used to create new behaviors in programs. **Lesson 7- Sprites:** Discuss the various information that programs must keep track of and how sprites are a way to keep track of that information. Learn how to assign each sprite an image. **Lesson 8- Sprite Properties:** Extend understanding of sprites by interacting with sprite properties. Reflect on the connections between properties and variables. **Lesson 9- Text:** Practice adding and placing text on the screen and controlling other text properties, such as size. **Lesson 10- Mini-Project- Captioned Scene:** Use the problem solving process as a model, define the scene to create, prepare by thinking of all of the code needed, try a plan in Game Lab, then reflect on what was created. **Lesson 11- The Draw Loop:** Learn how to combine the draw loop with random numbers to manipulate some simple animations. **Lesson 12- Sprite Movement:** Learn how to control sprite movement using a construct called the counter pattern and use the counter pattern to create various types of sprite movement. **Lesson 13- Mini- Project- Animation:** Combine different methods from previous lessons to create an animated scene. **Lesson 14- Conditionals:** Learn how the computer evaluates boolean expressions, and how they can be used to structure a program. **Lesson 15- Keyboard Inputs:** Learn to write programs that take keyboard input from the user to control the sprites on the screen.	• Respect Ability: Fighting Stigmas, Advancing Opportunities

Coding (CS Discoveries)

Lesson 16- Mouse Input: Expand understanding of conditional to include *else*, which allows for the computer to run a certain section of code when a condition is true, and a different section of code when it is not. Learn how to take mouse input. **Lesson 17- Project- Interactive Card:** Plan for and develop an interactive greeting card using all of the programming techniques learned to this point.	

Differentiation

*Please note: Teachers who have students with 504 plans that require curricular accommodations are to refer to Struggling and/or Special Needs Section for differentiation

High-Achieving Students	On Grade Level Students	Struggling Students	Special Needs/ELL
Higher order thinking questions Differentiation of pacing and activities Differentiation of learning strategies: visual, auditory, kinetic and cooperative Enrichment and extension Code.org challenge levels	Differentiation of learning strategies: visual, auditory, kinetic and cooperative Differentiating the lesson activities Lesson tutorials	Provide a highly structured, predictable learning environment Provide organizers Lessons designed to the style of learning that matches the student Cooperative Learning Positive reinforcement Lessons presentation available on google classroom Frequent check for understanding Break down task into manageable units One-on-one instruction Pair student with a high achieving student Provide video tutorials	Any student requiring further accommodations and/or modifications will have them individually listed in their 504 Plan or IEP. These might include, but are not limited to: breaking assignments into smaller tasks, giving directions through several channels (auditory, visual, kinesthetic, model), and/or small group instruction for reading/writing ELL supports should include, but are not limited to, the following:: Extended time Provide visual aids Repeated directions Differentiate based on proficiency Provide word banks Allow for translators, dictionaries

Coding (CS Discoveries)

Pacing Guide

Coding (CS Discoveries)	Resource	Standards
UNIT 1 Problem Solving and Computing (4 days)	**Lessons:** Lesson 1- Intro to Problem Solving: 2 days Lesson 2- The Problem Solving Process: 1 day Lesson 3- Exploring Problem Solving: 1 day	8.2.8.ED.2
UNIT 2 Web Development (18 days)	**Lessons:** Lesson 1- Exploring Webpages: 1 day Lesson 2- Intro to HTML: 1 day Lesson 3- Headings: 1 day Lesson 4- Mini-Project: HTML Webpage: 3 days Lesson 5- Digital Footprint: 1 day Lesson 6- Styling Text with CSS: 1 day Lesson 7- Mini-Project: Your Personal Style: 2 days Lesson 8- Intellectual Property: 1 day Lesson 9- Using Images: 1 day Lesson 10- Websites for Expression: 1 day Lesson 11- Styling Elements with CSS: 1 day Lesson 12- Your Web Page- Prepare: 1 day Lesson 13- Project- Personal Web Page: 3 days	8.1.8.AP.4 8.1.8.AP.5 8.1.8.AP.8 8.1.8.AP.9

Coding (CS Discoveries)

UNIT 3 Interactive Animations (18 days)	**Lessons:** Lesson 1- Programming for Entertainment: 1 day Lesson 2- Plotting Shapes: 1 day Lesson 3- Drawing in Game Lab: 1 day Lesson 4- Shapes and Parameters: 1 day Lesson 5- Variables: 1 day Lesson 6- Random Numbers: 1 day Lesson 7- Sprites: 1 day Lesson 8- Sprite Properties: 1 day Lesson 9- Text: 1 day Lesson 10- Mini-Project- Captioned Scene: 1 day Lesson 11- The Draw Loop: 1 day Lesson 12- Sprite Movement: 1 day Lesson 13- Mini- Project- Animation: 1 day Lesson 14- Conditionals: 1 day Lesson 15- Keyboard Inputs: 1 day Lesson 16- Mouse Input: 1 day Lesson 17- Project- Interactive Card: 2 days	8.1.8.AP.2 8.1.8AP.3 8.1.8.AP.4 8.1.8.AP.5 8.1.8.AP.8 8.1.8.AP.9