Unit 1: Introduction to coding

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

UNIT RATIONALE

Unit Rationale: The Introduction to No-Code Programs unit aims to introduce students to the fundamentals of coding and programming in a creative and accessible way. By using no-code platforms, students can explore the world of interactive project creation without the need for traditional coding knowledge. This unit encourages students to develop problem-solving skills, computational thinking, and creativity while gaining exposure to technology tools used in various industries.

ESSENTIAL QUESTIONS

1. What is coding, and how does it relate to computer programming?
2. How can we create interactive projects and applications without traditional coding?
3. What are the benefits and limitations of no-code programs?

STANDARDS

NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

SAVED

New Jersey (NJSLS) - Grades 6-8 - Computer Science and Design Thinking (2020)

8.1.8.CS.1:

Recommend improvements to computing devices in order to improve the ways users interact with the devices.

8.1.8.IC.1:

Compare the trade-offs associated with computing technologies that affect individual's everyday activities and career options.

8.2.8.ED.5:

Explain the need for optimization in a design process.

8.2.8.ED.7:

Design a product to address a real-world problem and document the iterative design process, including decisions made as a result of specific constraints and trade-offs (e.g., annotated sketches).

8.2.8.ITH.1:

Explain how the development and use of technology influences economic, political, social, and cultural issues.

8.2.8.NT.1:

Examine a malfunctioning tool, product, or system and propose solutions to the problem.

8.2.8.NT.4:

Explain how a product designed for a specific demand was modified to meet a new demand and led to a new product.

CS.6-8.8.1.8.CS.1	Recommend improvements to computing devices in order to improve the ways users interact with the devices.
CS.6-8.8.1.8.IC.1	Compare the trade-offs associated with computing technologies that affect individual's everyday activities and career options.
CS.6-8.8.2.8.ED.5	Explain the need for optimization in a design process.
CS.6-8.8.2.8.ED.7	Design a product to address a real-world problem and document the iterative design process, including decisions made as a result of specific constraints and trade-offs (e.g., annotated sketches).
CS.6-8.8.2.8.NT.1	Examine a malfunctioning tool, product, or system and propose solutions to the problem.
CS.6-8.8.2.8.NT.4	Explain how a product designed for a specific demand was modified to meet a new demand and led to a new product.
CS.6-8.8.2.8.ITH.1	Explain how the development and use of technology influences economic, political, social, and cultural issues.

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

CS.6-8.8.2.8.NT.4	Explain how a product designed for a specific demand was modified to meet a new demand and led to a new product.
CS.6-8.8.2.8.ETW.1	Illustrate how a product is upcycled into a new product and analyze the short- and long-term benefits and costs.
CS.6-8.8.2.8.ETW.2	Analyze the impact of modifying resources in a product or system (e.g., materials, energy, information, time, tools, people, capital).

NEW JERSEY STUDENT LEARNING STANDARDS: COMPUTER SCIENCE AND DESIGN THINKING

CS.6-8.8.1.8.CS.1	Recommend improvements to computing devices in order to improve the ways users interact with the devices.
CS.6-8.8.1.8.CS.2	Design a system that combines hardware and software components to process data.
CS.6-8.8.1.8.CS.3	Justify design decisions and explain potential system trade-offs.
CS.6-8.8.1.8.CS.4	Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.
CS.6-8.8.2.8.ED.2	Identify the steps in the design process that could be used to solve a problem.
CS.6-8.8.2.8.ED.3	Develop a proposal for a solution to a real-world problem that includes a model (e.g., physical prototype, graphical/technical sketch).

CS.6-8.8.2.8.ED.6	Analyze how trade-offs can impact the design of a product.

PRE-ASSESSMENTS

Students will complete a pre-assessment to determine their current knowledge about coding and what questions they may have.

INSTRUCTIONAL PLAN

MODULE 1

Unit 1: Introduction to No-Code Programs

Essential Questions:
1. What is coding, and how does it relate to computer programming?
2. How can we create interactive projects and applications without traditional coding?
3. What are the benefits and limitations of no-code programs?

Success Criteria:
1. Students will understand the basic concepts of coding and programming.
2. Students will demonstrate proficiency in using a no-code platform to create interactive projects.
3. Students will evaluate the effectiveness and efficiency of using no-code programs for various applications.

Learning Strategies:
1. Exploratory activities to familiarize students with no-code platforms.
2. Guided tutorials and step-by-step instructions for creating projects.
3. Collaborative projects to encourage peer learning and problem-solving.
4. Reflection and discussion on the advantages and challenges of using no-code programs.

Learning Intentions:
1. Develop an understanding of coding and programming concepts.
2. Gain proficiency in using a no-code platform to create interactive projects.
3. Foster creativity and problem-solving skills through project-based learning.

NJ Content Standards: 6.2.8.A.1, 6.2.8.A.2, 6.2.8.B.1, 6.2.8.B.2, 6.2.8.C.1, 6.2.8.C.2

Small Projects:
1. Create an interactive storybook using a no-code platform.
2. Design a simple game or quiz application without coding.
3. Develop a personal website or portfolio using a no-code platform.

Large Project:
Design and build a no-code application that solves a real-world problem or addresses a specific need.

REFLECTIONS

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

CS.6-8.8.1.8.CS.1	Recommend improvements to computing devices in order to improve the ways users interact with the devices.
CS.6-8.8.1.8.CS.2	Design a system that combines hardware and software components to process data.
CS.6-8.8.1.8.CS.3	Justify design decisions and explain potential system trade-offs.
CS.6-8.8.1.8.CS.4	Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.