01_Unit 1 The Design Process

Content Area: **Math**
Course(s):
Time Period: **Cycle**
Length: **7 lessons (22 lesson marking period cycle; 1 of 3 units)**
Status: **Published**

General Overview, Course Description or Course Philosophy

Computer Programming 8

Computer science and design thinking education prepares students to succeed in today's knowledge-based economy by providing equitable and expanded access to high-quality, standards-based computer science and technological design education. During 8th grade, students will focus on the core ideas of computing systems, networks, impacts of computing and data analysis, programming, engineering design, ethics and culture of technology, and the interaction and effects of technology with and on humans and the natural world. They do so by completing three specific units entitled "The Design Process", "Data and Society", and "Physical Computing". These follow a logical sequence and come AFTER the 7th-grade units (entitled "Problem-Solving and Computing", "Web Development", and "Interactive Animation and Games").

OBJECTIVES, ESSENTIAL QUESTIONS, ENDURING UNDERSTANDINGS

Unit Summary:

People design for enjoyment and to solve problems, extend human capabilities, satisfy needs and wants, and improve the human condition. Engineering Design, a systematic approach to creating solutions to technological problems and finding ways to meet people’s needs and desires, allows for the effective and efficient development of products and systems. Interaction of Technology and Humans concerns the ways society drives the improvement and creation of new technologies, and how technologies both serve and change society. Computer programmers develop algorithms, which are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems.

Essential Question(s):

- How can computer programmers use algorithms to help identify people's needs and desires in our society?
- How can technology be improved to better meet the needs of people?

- How can a computer program extend human capabilities and improve human conditions?

Enduring Understandings:

- Individuals design algorithms that are reusable in many situations.
- Algorithms that are readable are easier to follow, test, and debug.
- Programmers create variables to store data values of different types and perform appropriate operations on their values.
- Control structures are selected and combined in programs to solve more complex problems.
- Programs use procedures to organize code and hide implementation details. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.
- Individuals design and test solutions to identify problems taking into consideration the diverse needs of the users and the community.
- Economic, political, social, and cultural aspects of society drive development of new technological products, processes, and systems.
- Engineering design requirements and specifications involve making trade-offs between competing requirements and desired design features.

CONTENT AREA STANDARDS

CS.6-8.8.1.8.AP.1	Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
CS.6-8.8.1.8.AP.3	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
CS.6-8.8.1.8.AP.6	Refine a solution that meets users' needs by incorporating feedback from team members and users.
CS.6-8.8.1.8.AP.7	Design programs, incorporating existing code, media, and libraries, and give attribution.
CS.6-8.8.1.8.AP.8	Systematically test and refine programs using a range of test cases and users.
CS.6-8.8.1.8.AP.9	Document programs in order to make them easier to follow, test, and debug.
CS.6-8.8.2.8.ED.5	Explain the need for optimization in a design process.
CS.6-8.8.2.8.ED.6	Analyze how trade-offs can impact the design of a product.
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.ITH.1	Explain how the development and use of technology influences economic, political, social, and cultural issues.

RELATED STANDARDS (Technology, 21st Century Life & Careers, ELA Companion Standards are Required)

LA.L.7.1	Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.
LA.RI.7.10	By the end of the year read and comprehend literary nonfiction at grade level text-complexity or above, with scaffolding as needed.
WRK.K-12.P.4	Demonstrate creativity and innovation.
WRK.K-12.P.5	Utilize critical thinking to make sense of problems and persevere in solving them.
WRK.K-12.P.8	Use technology to enhance productivity increase collaboration and communicate effectively.
TECH.9.4.2.CT.1	Gather information about an issue, such as climate change, and collaboratively brainstorm ways to solve the problem (e.g., K-2-ETS1-1, 6.3.2.GeoGI.2).
TECH.9.4.2.CT.3	Use a variety of types of thinking to solve problems (e.g., inductive, deductive).

STUDENT LEARNING TARGETS

Refer to the 'Declarative Knowledge' and 'Procedural Knowledge sections.

Declarative Knowledge

Students will understand that:

- Programmers must be able to navigate broader social impacts of computing.
- Programmers must understand the needs of others while developing a solution to a problem; programmers must show empathy with and understanding of users.
- Computer science is a tool for social impact.

Procedural Knowledge

Students will be able to:

- Identify a need that they care about, prototype solutions both on paper and in App Lab, and test their solutions with real users (to get feedback and drive further iteration)
- explain the needs of others and be willing to incorporate feedback from presentations into their prototype of their App
- use code to program the App they designed on paper using all of the buttons and features in App Lab

EVIDENCE OF LEARNING

Refer to the 'Formative Assessments' and 'Summative Assessments' sections.

Formative Assessments

For this unit, formative assessments may include:

- observation
- one-on-one assistance
- questioning skills
- graphic organizers
- anecdotal notes
- exit tickets
- student interviews and check-ins

Summative Assessments

For this unit, summative assessments may include:

- graphic organizers
- homework, when applicable
- mini projects at the end of units
- culminating activities in the code.org units

RESOURCES (Instructional, Supplemental, Intervention Materials)

Code.org Website

All lessons and resources can be accessed via this website.

INTERDISCIPLINARY CONNECTIONS

English/Language Arts - implementation of conventions of Standard English

Technology/Multi-Media - audio/visual media analysis

Math - computations

Visual and Performing Arts- presentations on app lab and website design

Social Studies - ethical codes of components of technology

Science- computer science, physics

ACCOMMODATIONS & MODIFICATIONS FOR SUBGROUPS

See link to Accommodations & Modifications document in course folder.