Unit 4: Variables, Conditionals and Functions

Content Area: **Mathematics**
Course(s): **Generic Course**
Time Period: **Semester 1**
Length: **3 weeks**
Status: **Published**

Standards

	Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
	Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.
8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
	Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools.
8.1.12.AP.7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
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.

Essential Questions

What opportunities do large data sets provide for solving problems and creating knowledge?

How is cybersecurity impacting the ever-increasing number of Internet users?

How does cryptography work?

Enduring Understanding

- Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.
- To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
- The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow

programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

Knowledge and Skills

Students expand the types of apps they can create as they learn how to store information (variables), make decisions (conditionals), and better organize code (functions). Each programming topic is covered in a specific sequence of lessons that ask students to ‘Explore’ ideas through hands-on activities, ‘Investigate’ these ideas through guided code reading, ‘Practice’ with sample problems, and apply their understanding as they ‘Make’ a one-day scoped project. The entire unit concludes with a three-day open-ended project in which students must build an app that makes a recommendation about any topic they wish.

Transfer Goals

There is a differene between program development and program function.

Resources

1. Various YouTube videos that visually explain concepts and ideas.

2. Various widgets found on code.org.

3. Test banks created on Edulastic and code.org

4. Use of Google Classroom, Google Slides, Google Docs and Google Sheets