Unit #7: Parameters, Return, and Libraries

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

State Mandated Topics Addressed in this Unit

State Mandated Topics Addressed in this Unit	
N/A	N/A

Unit #7: Parameters, Return, and Libraries

Learning Objectives

- Artists explore the requirements for working with libraries and documentation.
- Artists will be designing functions and debugging complex programs that use parameters and return values
- Artists will design and build a library of functions around any topic they want.
- Artists will explore how functions with parameters and return values help further split programs into logical chunks of code.
- Artists will investigate how libraries are used in actual apps before they begin building libraries themselves.
- Artists will understand that libraries are higher level blocks in the toolbox that either make your code shorter, since library code does the heavy lifting, or more complex, because library code allows you to focus on bigger problems.
- Artists will understand the concept and high level context of parameters and returns.

Essential Skills

- Analyze the effects of changing resources when designing a specific product or system (e.g., materials, energy, tools, capital, labor).
- Clearly write documentation for functions in a library
- Correctly set up a parameter in a function
- Correctly set up a return value in a function
- Debug a library of functions based on testing and classmate feedback
- Debug library code to remove any errors
- Debug programs that use functions with parameters and return

- Design the API for a library of functions, including the function names, purpose, and parameters, and types of values each function will return
- 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)
- Explain the benefits of using a function with a parameter or return value in the context of a specific program
- Explain the process of preparing a function to be added to a library
- Explain the purpose and functionality of a function they developed
- Explain the purpose of libraries as a way to simplify programs, allow for code reuse, and enable collaboration.
- Explain two different calls to a function they developed
- Identify situations in which a function with a parameter or return value would be necessary
- Identify the use of a library within a program.
- Modify programs that use functions with parameters and return
- Provide feedback to their classmate about a library they designed
- Read library code documentation in order to select the proper functions in the library to develop an app
- Remove specifics from a function so that it can be used in a variety of situations
- Select a theme for a library of functions
- Test functions designed to be used in a library using different input values
- Test the functions in libraries in order to understand their behavior
- Use appropriate vocabulary to describe libraries
- Use appropriate vocabulary to describe parameters and return values.
- Use the modulus operator in a program
- Write comments to explain the function purpose, parameters, and return values
- Write functions with parameters and return values that meet a set of specified requirements
- Write tests for functions with a library that they designed

Standards

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.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
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.7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
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.9-12.8.2.12.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.ED.6	Analyze the effects of changing resources when designing a specific product or system (e.g., materials, energy, tools, capital, labor).

Instructional Tasks/Activities

- Classroom Discussions
- Debugging
- Exploration/Investigation/App Lab - Design Mode Activities
- Formative Assessments
- Journaling
- Pair Programming
- Peer Feedback
- Project Make A Library - modified from code.org
- Worksheets

Assessment Procedure

- Classroom Total Participation Technique
- Classwork
- DBQ
- electronic active responders
- Essay
- Exit Ticket/Entrance Ticket/Do Now
- identify the error problems
- Journal / Student Reflection
- Kahoot
- Other named in lesson
- Peer Review
- Performance
- Problem Correction

- Project
- Quiz
- response and analysis questions
- Rubric
- Teacher Collected Data
- Test
- Worksheet

Recommended Technology Activities

- App Lab
- Appropriate Content Specific Online Resource
- Gimkit
- GoGuardian
- Google Classroom
- Google Docs
- Google Slides
- Google Slides
- Kahoot
- MagicSchool AI
- Other- Specified in Lesson
- Quiziz
- Screencastify

Accommodations & Modifications & Differentiation

Accommodations and Modifications should be used to meet individual needs. Their IEP and 504 plans should be used in addition to the following suggestions.

Gifted and Talented

- Compare & Contrast
- Conferencing
- Debates
- Jigsaw
- Peer Partner Learning
- Problem Solving
- Structured Controversy

- Think, Pair, Share
- Tutorial Groups

Instruction/Materials

- alter format of materials (type/highlight, etc.)
- color code materials
- eliminate answers
- extended time
- extended time
- large print
- modified quiz
- modified test
- Modify Assignments as Needed
- Modify/Repeat/Model directions
- necessary assignments only
- Other (specify in plans)
- other- named in lesson
- provide assistance and cues for transitions
- provide daily assignment list
- read class materials orally
- reduce work load
- shorten assignments
- study guide/outline
- utilize multi-sensory modes to reinforce instruction

Environment

- alter physical room environment
- assign peer tutors/work buddies/note takers
- assign preferential seating
- individualized instruction/small group
- modify student schedule (Describe)
- other- please specify in plans
- provide desktop list/formula

Honors Modifications

Resources

- code.org
- Internet Simulator