Unit 7 Lesson 1

Parameters and Return Explore

Resources

Name(s)____ Period _______ Date ____________________

Activity Guide - Function Houses

Instructions:

Challenge #1: Time to create your own function with parameters!

- Create a function envelope. Put the name of the function on the envelope.
- On a scrap piece of paper, write your function. It can be written in pseudocode. Describe how to draw a simple house on graph paper.
    - Think about the parameters. How could your function be used to draw different types or sizes of houses? Give each parameter a unique name.
    - You can have as many parameters as you want. When the parameter is referenced in the function, put a box around it to draw attention.
    - Put the function inside the envelope.
- Turn to the front of the envelope. Put boxes on the envelope for each parameter and put the name of the parameter in the box.
- Choose arguments for each parameter. Write them on small sticky notes and stick them in the correct boxes on the envelope.
- Pass the envelope to a partner. The partner will draw the house following the instructions in the function, using the parameters.
- Once they've finished, check to see if the drawing is as expected.
- Choose new arguments. Replace the old ones on the envelope with the new ones and "call" the function again by passing it to a partner. The partner then draws the new house.

Challenge #2: Create a cost calculator function for building the house you created a function for earlier.

- Create a function envelope. Put the name of the function on the envelope.
- On a scrap piece of paper, write your function. It can be written in pseudocode. Create a cost calculator.
    - Think about the parameters. How would these impact the cost?
    - Add a return at the end of your function that returns the total cost.
- Turn to the front of the envelope. Put boxes on the envelope for each parameter and put the name of the parameter in the box.
- Choose arguments for each parameter. Write them on small sticky notes and stick them in the correct boxes on the envelope.
- Pass the envelope to a partner. The partner will calculate the cost of building the house using the parameters.
- Once they've finished, the partner should pass back or **return** a sticky note with the value of the total cost.
- Then choose new arguments. Replace the old ones on the envelope with the new ones and "call" the function again by passing it to a partner. The partner will now run the calculations again and complete the **return**. Partners should check to see if the **return** value is correct!

Unit 7 Lesson 2

Parameters and Return Investigate

Resources

Unit 7 Lesson 3

Parameters and Return Practice

Resources

Unit 7 Lesson 4

Parameters and Return Make

Resources

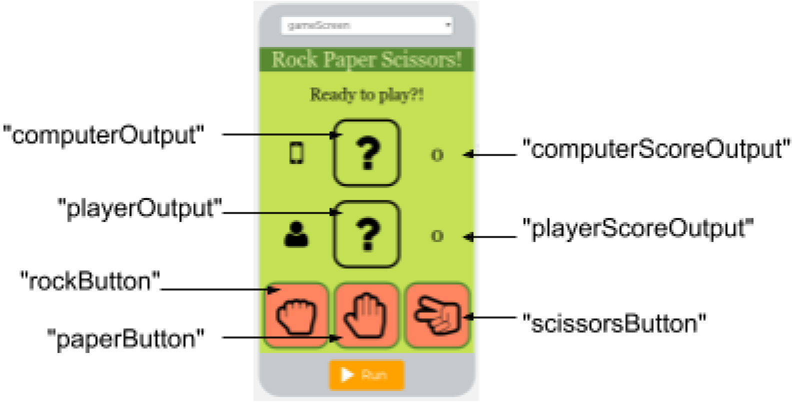
Name(s)____ Period _______ Date ____________________

Activity Guide - Parameters and Return Make

Step 1 - Try the app

Try playing the game. Pay attention to:

- What each button does
- How the screen gets updated after clicking each button



Step 2 - Plan

This Make project is a little different than other ones. Instead of writing all of the code yourself, most of it has already been written for you. There are just three functions at the bottom that have been designed and have comments explaining how they should work but the code for each function is incomplete or only works for some inputs. Read the comments for these three functions carefully. You can also quickly read the rest of the program if you like.

In the table below record any notes for how you will build each function

	Notes for Building this Function
findIcon()	
randomChoose()	
decideWinner()	

Step 3 - Write Your Code

- Write the code for the app, using your plan above and the comments provided in Code Studio to help. You can also check the Help & Tips tab for Programming Patterns you can use
- Step You Can Follow
  - Write the code for findIcon() first. Notice that the list of possible icons is available on Line 2
  - Test your code. Once your program is working you should see the correct icon show up on the game board when you pick rock, paper or scissors.
  - Write the code to make the randomChoose() function work as described. You should only be working inside that function
  - Test your code. Once your program is working you should see the computer player is able to pick randomly between rock, paper, and scissors.
  - Write the code to make the decideWinner() function work as described. You should only be working inside that function
  - Test your code. Once this function is done your app should be completely working and correctly updating the score on the screen.
  - For reference: Rock beats Scissors, Paper beats Rock, Scissors beats Paper
  - Add comments to each of your functions explaining how you wrote the code.
  - Test the app to make sure it works for every combination of rock, paper, and scissors.
- Extension Ideas
  - Have the game end when one player gets to 10 wins
  - Keep track of the current streak of wins in the game

Step 4 - Submit

Before your submit check the rubric below to make sure your program

Category	Extensive Evidence	Convincing Evidence	Limited Evidence	No Evidence
findIcon Function	The function returns the correct values for all input values.	The function returns the correct values for most input values.	The function returns the correct values for some input values.	The function does not return correct values for any input values.
randomChoose Function	The function returns the correct values for all input values.	The function returns the correct values for most input values.	The function returns the correct values for some input values.	The function does not return correct values for any input values.
decideWinner Function	The function returns the correct values for all combinations of inputs.	The function returns the correct values for most combinations of inputs.	The function returns the correct values for some combination of inputs.	The function does not return correct values for any combination of inputs
Code runs without errors.	No errors are present in the required code.	Some errors are present in the required code.	Many errors are present in the required code.	The code does not run.
Coding Comments	Comments are used to correctly explain the purpose and functionality of all functions.	Comments are used to explain the purpose and functionality of most functions.	Comments are present, but are not used to explain the purpose or functionality of any functions.	Comments are not present.

Unit 7 Lesson 5

Libraries Explore

Resources

Unit 7 Lesson 6

Libraries Investigate

Resources

Unit 7 Lesson 7

Libraries Practice

Resources

Unit 7 Lesson 8

Project - Make a Library Part 1

Resources

Name(s)____ Period ______ Date ___________________

Project - Make a Library

Project Description

Libraries let you build functions that you and others can use to help build more powerful apps in lots of situations. As the designer of a library you not only need to know how to program, you also need to think about the many different kinds of programs other people might build.

For this project you will design and build a library of functions around any topic you want. You will have an opportunity to exchange feedback with another group about how you might use their library to design an app. Finally you will answer a few questions about the library you designed.

You will submit

- A link to your library of functions
- This completed project guide

Library Requirements

- Your library must contain two or more functions
- At least one function must include
  - A loop
  - An if-statement
  - One or more parameters
  - Return

Step 1 - Brainstorm

Your library can be about any topic. If you need some ideas try the list below

- **Strings**
  - Capitalize: Capitalize first letter of each word in a sentence (long string)
  - Trim: Remove spaces from beginning and end of a string
  - Remove Letter: removes a given letter from a string
  - Anything else you'd want to do with a string

- **Lists**
  - Maximum: Find the maximum value in a list
  - Minimum: Find the minimum value in a list
  - Average: Find the average value in a list
  - Count: Count how many times a given value appears in a list
  - Combine: Join two lists together in one longer, larger list
  - Filter Numbers: Keep numbers in a list greater than or less than a provided value
  - Filter Letters: Keep strings in a list that begin with a given letter
  - Numbered List: Turn a list into a string with each item numbered and appearing on a different line
  - Top 3: Return the three biggest numbers in a list, in sorted order (This can be a little tricky)
  - Unique: Return a list of each unique item that appears in (This can be a little tricky)
  - Sort: Return the list in sorted order (This can be a little tricky)
  - Anything else you'd want to do with a list

- **Dataset**
  - Choose a dataset and build a function that will help a user access or summarize specific information within that dataset

- **Something else**
  - As long as your library does something interesting and valuable

Step 2 - Design

After you've brainstormed the focus of your app, fill in the table below with the list of functions you intend to build.

Function Name *Write the name of each function*	**Description, Parameters, Return** *Write the comments for this function*	**Loop, If-statement, Param, Return?** *At least one function must include all 4 features listed above*
maximum(list)	*// Takes a list and returns the largest value that appears in the list* *// list {list} - the list of items* *// return {number} - the largest number in the list*	*Yes this function should include all of these features.*

Step 3 - Build

Program your library. Make sure you do the following.

- Start by building the function that will include all four required features (param, return, if-statement, loop)
- Debug your code as you write it
- Start testing your code early with sample inputs

Step 4 - Test

Write test cases for each of the functions in your library to make sure they return the expected values. Remember that you should include.

- Inputs (arguments) that will result in your functions behaving differently or returning different values
- Inputs (arguments) just before, at, and after cut offs if your conditionals. For example if your code includes the statement (value < 2) then try inputs where value is 1, 2, and 3 to see if the algorithm always behaves as you expect.

Step 5 - Share and Feedback

Send your library to a classmate and have them send it to you. You shouldn't need to explain anything about how your library works or what it is for. The documentation should be good enough.

Your reviewer should fill in the information below.

Reviewer Name: ____

	Yes	Kind Of	No
Clear: I can easily understand the overall purpose of the library and each function within it.			
Error-Free: Each function in the library works as expected.			
Useful: I can think of situations where I would want to use this library			

I like: Give feedback on at least one thing you like in the library

____

____

I wish: Give one problem or limitation of the library

____

____

What if: Give one idea for how to improve the library

____

____

Step 6 - Improve

Based on the feedback above make final improvements to your library.

Step 7 - Acknowledge Collaborators / Sources

It is important to acknowledge any code you write that was developed collaboratively or that you received from another source. In the table below list work you completed with a collaborative partner or got from another source

Name / Source	Lines of Code / Portions of Program Contributed

Step 8 - Free Response

1. Copy and paste a function from your library that:
    a. uses one or more parameters that have an effect on the functionality of the procedure; and
    b. implement an algorithm that includes sequencing, selection and iteration.

Paste the code of your procedure here

2. Provide a written response that:
    a. Describes in general what the identified procedure does and how it contributes to the overall functionality of the program
    b. Explains in detailed steps how the algorithm implemented in the identified procedure works. Your explanation must be detailed enough for someone else to recreate it.

Write your response here

3. Provide a written response that:
    a. Describes two calls to the selected function. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
    b. Describes what condition(s) is being tested by each call to the procedure; and
    c. identifies the result of each call.

	Call 1	**Call 2**
Arguments What arguments are passed?		
Conditions Checked What conditional statement is checked to make different segments of the function run		
Results What will happen / be returned by this function call?		

Scoring Guidelines

Written Responses (individual) - In the Written Response the student...	
Responses	**Point**
Selected procedure includes a parameter that impacts the functionality of the procedure	**/1**
Selected procedure includes sequencing, selection, and iteration	**/1**
Response 2 explains what the selected procedure does	**/1**
Response 2 explains how the selected procedure accomplishes its task	**/1**
Response 3 Call 1: the response indicates the arguments that are passed, the specific condition that is checked, and the result of the call.	**/1**
Response 3 Call 2: the response indicates the arguments that are passed, the specific condition that is checked, and the result of the call.	**/1**
Program Code	
Library includes two or more functions that accomplish related tasks	**/2**
All functions include comments that explain the purpose of the function and the parameters	**/2**
All functions include tests that demonstrate the code working as expected	**/2**
Library code runs without errors	**/1**
Project Guide	
Project guide is complete	**/1**
Student provided clear and actionable feedback to a classmate on their project guide	**/1**
Total	**/15**

Unit 7 Lesson 9

Project - Make a Library Part 2

Resources

Name(s)____ Period ______ Date ___________________

Project - Make a Library

Project Description

Libraries let you build functions that you and others can use to help build more powerful apps in lots of situations. As the designer of a library you not only need to know how to program, you also need to think about the many different kinds of programs other people might build.

For this project you will design and build a library of functions around any topic you want. You will have an opportunity to exchange feedback with another group about how you might use their library to design an app. Finally you will answer a few questions about the library you designed.

You will submit

- A link to your library of functions
- This completed project guide

Library Requirements

- Your library must contain two or more functions
- At least one function must include
    - A loop
    - An if-statement
    - One or more parameters
    - Return

Step 1 - Brainstorm

Your library can be about any topic. If you need some ideas try the list below

- **Strings**
    - Capitalize: Capitalize first letter of each word in a sentence (long string)
    - Trim: Remove spaces from beginning and end of a string
    - Remove Letter: removes a given letter from a string
    - Anything else you'd want to do with a string

- **Lists**
    - Maximum: Find the maximum value in a list
    - Minimum: Find the minimum value in a list
    - Average: Find the average value in a list
    - Count: Count how many times a given value appears in a list
    - Combine: Join two lists together in one longer, larger list
    - Filter Numbers: Keep numbers in a list greater than or less than a provided value
    - Filter Letters: Keep strings in a list that begin with a given letter
    - Numbered List: Turn a list into a string with each item numbered and appearing on a different line
    - Top 3: Return the three biggest numbers in a list, in sorted order (This can be a little tricky)
    - Unique: Return a list of each unique item that appears in (This can be a little tricky)
    - Sort: Return the list in sorted order (This can be a little tricky)
    - Anything else you'd want to do with a list

- **Dataset**
    - Choose a dataset and build a function that will help a user access or summarize specific information within that dataset

- **Something else**
    - As long as your library does something interesting and valuable

Step 2 - Design

After you've brainstormed the focus of your app, fill in the table below with the list of functions you intend to build.

Function Name *Write the name of each function*	**Description, Parameters, Return** *Write the comments for this function*	**Loop, If-statement, Param, Return?** *At least one function must include all 4 features listed above*
maximum(list)	*// Takes a list and returns the largest value that appears in the list* *// list {list} - the list of items* *// return {number} - the largest number in the list*	*Yes this function should include all of these features.*

Step 3 - Build

Program your library. Make sure you do the following.

- Start by building the function that will include all four required features (param, return, if-statement, loop)
- Debug your code as you write it
- Start testing your code early with sample inputs

Step 4 - Test

Write test cases for each of the functions in your library to make sure they return the expected values. Remember that you should include.

- Inputs (arguments) that will result in your functions behaving differently or returning different values
- Inputs (arguments) just before, at, and after cut offs if your conditionals. For example if your code includes the statement (value < 2) then try inputs where value is 1, 2, and 3 to see if the algorithm always behaves as you expect.

Step 5 - Share and Feedback

Send your library to a classmate and have them send it to you. You shouldn't need to explain anything about how your library works or what it is for. The documentation should be good enough.

Your reviewer should fill in the information below.

Reviewer Name: ____

	Yes	Kind Of	No
Clear: I can easily understand the overall purpose of the library and each function within it.			
Error-Free: Each function in the library works as expected.			
Useful: I can think of situations where I would want to use this library			

I like: Give feedback on at least one thing you like in the library

____

____

I wish: Give one problem or limitation of the library

____

____

What if: Give one idea for how to improve the library

____

____

Step 6 - Improve

Based on the feedback above make final improvements to your library.

Step 7 - Acknowledge Collaborators / Sources

It is important to acknowledge any code you write that was developed collaboratively or that you received from another source. In the table below list work you completed with a collaborative partner or got from another source

Name / Source	Lines of Code / Portions of Program Contributed

Step 8 - Free Response

1. Copy and paste a function from your library that:
   a. uses one or more parameters that have an effect on the functionality of the procedure; and
   b. implement an algorithm that includes sequencing, selection and iteration.

Paste the code of your procedure here

2. Provide a written response that:
   a. Describes in general what the identified procedure does and how it contributes to the overall functionality of the program
   b. Explains in detailed steps how the algorithm implemented in the identified procedure works. Your explanation must be detailed enough for someone else to recreate it.

Write your response here

3. Provide a written response that:
   a. Describes two calls to the selected function. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
   b. Describes what condition(s) is being tested by each call to the procedure; and
   c. identifies the result of each call.

	Call 1	**Call 2**
Arguments What arguments are passed?		
Conditions Checked What conditional statement is checked to make different segments of the function run		
Results What will happen / be returned by this function call?		

Scoring Guidelines

Written Responses (individual) - In the Written Response the student...	
Responses	**Point**
Selected procedure includes a parameter that impacts the functionality of the procedure	**/1**
Selected procedure includes sequencing, selection, and iteration	**/1**
Response 2 explains what the selected procedure does	**/1**
Response 2 explains how the selected procedure accomplishes its task	**/1**
Response 3 Call 1: the response indicates the arguments that are passed, the specific condition that is checked, and the result of the call.	**/1**
Response 3 Call 2: the response indicates the arguments that are passed, the specific condition that is checked, and the result of the call.	**/1**
Program Code	
Library includes two or more functions that accomplish related tasks	**/2**
All functions include comments that explain the purpose of the function and the parameters	**/2**
All functions include tests that demonstrate the code working as expected	**/2**
Library code runs without errors	**/1**
Project Guide	
Project guide is complete	**/1**
Student provided clear and actionable feedback to a classmate on their project guide	**/1**
Total	**/15**

Unit 7 Lesson 10

Project - Make a Library Part 3

Resources

Name(s)____ Period ______ Date ____________________

Project - Make a Library

Project Description

Libraries let you build functions that you and others can use to help build more powerful apps in lots of situations. As the designer of a library you not only need to know how to program, you also need to think about the many different kinds of programs other people might build.

For this project you will design and build a library of functions around any topic you want. You will have an opportunity to exchange feedback with another group about how you might use their library to design an app. Finally you will answer a few questions about the library you designed.

You will submit

- A link to your library of functions
- This completed project guide

Library Requirements

- Your library must contain two or more functions
- At least one function must include
    - A loop
    - An if-statement
    - One or more parameters
    - Return

Step 1 - Brainstorm

Your library can be about any topic. If you need some ideas try the list below

- **Strings**
    - Capitalize: Capitalize first letter of each word in a sentence (long string)
    - Trim: Remove spaces from beginning and end of a string
    - Remove Letter: removes a given letter from a string
    - Anything else you'd want to do with a string

- **Lists**
    - Maximum: Find the maximum value in a list
    - Minimum: Find the minimum value in a list
    - Average: Find the average value in a list
    - Count: Count how many times a given value appears in a list
    - Combine: Join two lists together in one longer, larger list
    - Filter Numbers: Keep numbers in a list greater than or less than a provided value
    - Filter Letters: Keep strings in a list that begin with a given letter
    - Numbered List: Turn a list into a string with each item numbered and appearing on a different line
    - Top 3: Return the three biggest numbers in a list, in sorted order (This can be a little tricky)
    - Unique: Return a list of each unique item that appears in (This can be a little tricky)
    - Sort: Return the list in sorted order (This can be a little tricky)
    - Anything else you'd want to do with a list

- **Dataset**
    - Choose a dataset and build a function that will help a user access or summarize specific information within that dataset

- **Something else**
    - As long as your library does something interesting and valuable

Step 2 - Design

After you've brainstormed the focus of your app, fill in the table below with the list of functions you intend to build.

Function Name *Write the name of each function*	**Description, Parameters, Return** *Write the comments for this function*	**Loop, If-statement, Param, Return?** *At least one function must include all 4 features listed above*
maximum(list)	*// Takes a list and returns the largest value that appears in the list* *// list {list} - the list of items* *// return {number} - the largest number in the list*	*Yes this function should include all of these features.*

Step 3 - Build

Program your library. Make sure you do the following.

- Start by building the function that will include all four required features (param, return, if-statement, loop)
- Debug your code as you write it
- Start testing your code early with sample inputs

Step 4 - Test

Write test cases for each of the functions in your library to make sure they return the expected values. Remember that you should include.

- Inputs (arguments) that will result in your functions behaving differently or returning different values
- Inputs (arguments) just before, at, and after cut offs if your conditionals. For example if your code includes the statement (value < 2) then try inputs where value is 1, 2, and 3 to see if the algorithm always behaves as you expect.

Step 5 - Share and Feedback

Send your library to a classmate and have them send it to you. You shouldn't need to explain anything about how your library works or what it is for. The documentation should be good enough.

Your reviewer should fill in the information below.

Reviewer Name: ____

	Yes	Kind Of	No
Clear: I can easily understand the overall purpose of the library and each function within it.			
Error-Free: Each function in the library works as expected.			
Useful: I can think of situations where I would want to use this library			

I like: Give feedback on at least one thing you like in the library

____

____

I wish: Give one problem or limitation of the library

____

____

What if: Give one idea for how to improve the library

____

____

Step 6 - Improve

Based on the feedback above make final improvements to your library.

Step 7 - Acknowledge Collaborators / Sources

It is important to acknowledge any code you write that was developed collaboratively or that you received from another source. In the table below list work you completed with a collaborative partner or got from another source

Name / Source	Lines of Code / Portions of Program Contributed

Step 8 - Free Response

1. Copy and paste a function from your library that:
    a. uses one or more parameters that have an effect on the functionality of the procedure; and
    b. implement an algorithm that includes sequencing, selection and iteration.

Paste the code of your procedure here

2. Provide a written response that:
    a. Describes in general what the identified procedure does and how it contributes to the overall functionality of the program
    b. Explains in detailed steps how the algorithm implemented in the identified procedure works. Your explanation must be detailed enough for someone else to recreate it.

Write your response here

3. Provide a written response that:
    a. Describes two calls to the selected function. Each call must pass different arguments that cause a different segment of code in the algorithm to execute; and
    b. Describes what condition(s) is being tested by each call to the procedure; and
    c. identifies the result of each call.

	Call 1	**Call 2**
Arguments What arguments are passed?		
Conditions Checked What conditional statement is checked to make different segments of the function run		
Results What will happen / be returned by this function call?		

Scoring Guidelines

Written Responses (individual) - In the Written Response the student...	
Responses	**Point**
Selected procedure includes a parameter that impacts the functionality of the procedure	**/1**
Selected procedure includes sequencing, selection, and iteration	**/1**
Response 2 explains what the selected procedure does	**/1**
Response 2 explains how the selected procedure accomplishes its task	**/1**
Response 3 Call 1: the response indicates the arguments that are passed, the specific condition that is checked, and the result of the call.	**/1**
Response 3 Call 2: the response indicates the arguments that are passed, the specific condition that is checked, and the result of the call.	**/1**
Program Code	
Library includes two or more functions that accomplish related tasks	**/2**
All functions include comments that explain the purpose of the function and the parameters	**/2**
All functions include tests that demonstrate the code working as expected	**/2**
Library code runs without errors	**/1**
Project Guide	
Project guide is complete	**/1**
Student provided clear and actionable feedback to a classmate on their project guide	**/1**
Total	**/15**

Unit 7 Lesson 11

Assessment Day

Resources