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.*
find(list, word)	*// Checks to see if a list contains the given word* *// list [list] - the list of words* *// words [String] - the target word to find* *// return - if the word is in the list or not*	Yes
getFirst(list)	*// Grabs the first word from a list of words* *// list [list] - the list of words* *// return - the first word in the list*	No
getLast(list)	*// Grabs the last word from a list of words* *// list [list] - the list of words* *// return - the last word in the list*	No
gradIndex(list, word)	*// Grabs the index of where a word is* *// at within the list* *// list {list} - a list of words* *// word {String} - the word we are looking for the index of* *// return {number or String} - either an index or a message saying* *// the word cannot be found in the list*	Yes

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: ____Ryan____

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

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

I like how useful the finds function is. I can really see how that would be used and it seems so helpful.

I wish: Give one problem or limitation of the library

grabFirst and grabLast don't seem to be that useful. In any case, the function is the same as doing list[0] or list[list.length] which makes it so that these function aren't as useful as the first one that you have.

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

Maybe instead of having a grabFirst and grabLast you could have a function like grabIndex that grabs the index of where a word is in the list if it is in the list.

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
Ryan	Helped us create the findIndex() function

Step 8 - Free Response

1. Copy and paste a function from your library that:
    a. contains and 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

```
function finds(list, word) {
  for(var i = 0; i < list.length; i++) {
    if(list[i] == word) {
      return true;
    }
  }
  return false;
}
```

2. Provide a written response that:
    a. describes what the selected function does
    b. explains how the algorithm implemented in the selected function accomplishes its task.

Write your response here

The function checks to see if a word is already in a list.

What the function will do is it will loop through each element of the list in a for loop, and for each element, list[i], it has an if check to see if that element equals the input word the user is looking for. If so, the function will return true once it finds the word. However, if we finish the loop, that means the word was never found, so we return false.

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?
	list = [“Hello”, “I”, “am”, “a”, “human”]
	word = “human”
	finds(list, word)
		list = [“Hello”, “I”, “am”, “a”, “human”]
		word2 = “no”
		finds(list, word2)

Conditions Checked What conditional statement is checked to make different segments of the function run	The conditional statement checked is `if(list[i] == word) and whether that is true or not`	
Results What will happen / be returned by this function call?	In this case, “human” is in the list so it will enter the if check and return true	In this case, “no” is not in the list so it will never enter the if check and return hit the end of the loop and the return false statement at the end

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