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 library, 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.

## 

## Note: Once you are done testing your functions, make sure to comment out any tests you have written into your code to test your library functions before sending your library to a classmate in Step 5.

## 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 definition from your library that:
   1. uses one or more parameters that have an effect on the functionality of the procedure; and
   2. implement an algorithm that includes sequencing, selection and iteration.

| *Paste the code of your procedure here* |
| --- |

1. Provide a written response that:
   1. 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* |
| --- |

1. Provide a written response that:
   1. 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
   2. Describes what condition(s) is being tested by each call to the procedure; and
   3. identifies the result of each call.

|  | **Call 1** | **Call 2** |
| --- | --- | --- |
| **Arguments**  What arguments are passed?  (*inputs*) |  |  |
| **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?  (*outputs*) |  |  |

## 

## 

## 

## 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 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** | **/14** |