Applied Computer Science

Assignment 2

ACS-2909 Internet Programming

Due Date: Tuesday March 3, 2020 11:59 pm
Total Marks: 24

Create an HTML file which loads a separate JavaScript file which performs this functionality.

Part A

Use JavaScript to generate an *array* of 100 *objects*. These objects should have two properties (name and value). The name property should be your first name *(sorry no spaces)* concatenated with a zero-based counter, and the value should be the counter.

Save the array to a variable called `original` and output it to the console.

Example: My name is Paulo, so my code would generate this *array*:

```
[
      {'name': "Paulo0", 'value':0},
      {'name': "Paulo1", 'value':1},
                …
      {'name': "Paulo99", 'value':99}
]
```

Part B

Create a *function* called `toUpperTimesFive(original_array)` that will accept the `original` array and generate a new array that converts every *object* so that the name is all uppercase, and the values are the *original multiplied by 5.* The function must return the array.

Store the result into a variable called `upper` and output it to the console.

Part C

Create a *function* called `toLowerTimesThree(original_array)` that will accept the original array and generate a new array that converts every *object* so that the name is all lowercase and the values are the *original multiplied by 3.* The function must return the array.

Store the result into a variable called `lower` and output it to the console.

Part D

Create a *function* called `divisibles(upper, lower)` that accepts the two arrays from the previous steps. It will generate a new array that is formatted like `upper` but each object has an additional *key* called `found`. The `found` key is an array of all *objects* in `lower` that evenly divide into the upper value for that object, ignoring zero. The function must return the array.

Store the result into a variable called `final_result` and output it to the console.

Note: The other values should not be modified as a result of any processing. Arrays and objects pass by reference, so be cautious of how you use them.

Partial Output Example: For the `upper` value of “15”, the lower objects with of “3” and “15” will divide evenly into it. An array of those `lower` objects would be stored in *found*.

```
{
      'name': "PAULO3",
      'value': 15,
      'found':[
                  { 'name': "paulo1", 'value': 3},
                  { 'name': "paulo5", 'value': 15}
               ]
}
```

Hand In Instructions:

Zip all files into a single archive named ***StudentNumber*_Assignment2.zip**. Submit the zip file to the marker at 2909-051@acs.uwinnipeg.ca.

You *must* use your uwinnipeg email address!

Please allow yourself time to package the file and send it. Anything *received* after the deadline is considered late.