ACS Applied Computer Science

THE UNIVERSITY OF WINNIPEG

Applied Computer Science

ACS-2909-050
Internet Programming

Fall 2018

Assignment 4

Due Date: Nov 28th, 2018 11:59 pm **Total Marks: 24**

Motification

The goal of this assignment is use web workers, JSONP and websockets. You will use the provided HTML files located at https://courses.acs.uwinnipeg.ca/2909-050/assignments/A4.zip.

1. (12 marks) We want to process a large array of numbers, checking how many of the numbers in the array are perfect squares (i.e. the square root is an integer). We will use 10 Web Workers to process the array with 10 million numbers. Initially, each web worker will be given 1000 numbers from the array. Once a web worker is finished processing their "block" of 1000 numbers, that web worker should be given another block of 1000 numbers, OR they should terminate the web worker if there are no more "blocks" left.
    - All JS Code should reside in the *// JS CODE START* section
    - On load of the application, 10 web workers should be initialized and passed in blocks of 1000 numbers from the variable called ARRAY. i.e. Web Worker 1 should get 0 to 999 numbers, Webworker 2 should get 1000 to 1999 numbers, etc.

Assignment 4 (cont'd)

- The Web Worker code should count up all the numbers that are perfect squares (eg. 16 is a perfect square with a result of 4, 18 is not)
- The Web Worker will then return the count back to the main thread which will keep a total of all the perfect squares
- When a web worker finishes processing their block of 1000 numbers, it should be given another block of 1000 numbers. If there are no more blocks to process, the web worker should terminate
- Once all numbers have been processed, the main thread execute the given method

```
outputCount(count);
```

2. (12 marks) We have a service that has a list of 10 people and each person is repeatedly saying different phrases. We want to build an application that will initially call a JSONP service which will return a specific person object. The person object has a *name, title* and *id* property. Once retrieved, we want to open a web socket request to our service. On each message retrieved, we only want to print out phrases where the messages ID matches the ID of the person we initially retrieved.

- On load, make a JSONP call to our webservice. The URL for our JSONP service can be retrieved by calling the provided method:

```
httpUrl()
```

  - The query parameter *callback* is required for the JSONP call
- Once the JSONP request returns, write out *title* and *name* to the element with an id of *person.* The element should contain data that looks similar to

```
Some Job Title Suzy Jones
```

- After the JSONP call is completed, open a websocket request. The URL for the websocket service can be retrieved by calling the provided method:

```
wsUrl()
```

Assignment 4 (cont'd)

- Note that a "send" is necessary to initiate websocket responses
- The messages retrieved will have the properties *id, words* and *date.*
- Only the messages that have the same *id* property as the person retrieved from the JSONP call should be accepted
- Each message that has a matching *id*, we want to output to our unordered list with an id of *wordsSpoken.*
- The format for the list item should have the *words* followed by the *date* in parenthesis. For example,

```
Some Words Retrieved (2018-11-13T10:38:04)
```

Hand In Instructions:

Zip all files into a single archive named *StudentNumber_Assignment4.zip*. Submit the zip file to Nicole Van Hove at *vanhove-n@webmail.uwinnipeg.ca*.