**FOP2 Mock Practical Test**

DIPLOMA IN INFORMATION TECHNOLOGY

FIRST YEAR FULL TIME

**Fundamental of Programming 2**

**Time Allowed:** 1.5 Hour

Admission Number: p2322993 Class: DIT/FT/1B/04

Name: Angie Wu En Qi

# Instructions to Candidates

1. This paper consists of **9** pages (inclusive of the cover page and appendix).
2. Answer ALL questions.

* This is a **RESTRICTED OPEN BOOK** Test. You may refer to source codes on your laptop, access the module materials on the BrightSpace and online material related to JavaScript coding.
* **Do not use any communication software or access any websites that provide online e-communication (e.g Web WhatApp ,etc). Any form of communication online or offline is prohibited. Anyone caught doing so will be considered cheating which may result in failing the module or all the modules, suspension or expulsion.**
* **You are not allowed to use Generative AI tool (example: ChatGPT co-pilot, Visual Studio Code Co-Pilot or any form of Generative AI plug-in) for the Mid-Semester Test. Anyone caught doing so will be considered cheating which may result in failing the module or all the modules, suspension or expulsion.**
* Write your **admission number, name and class** at the beginning of every program as **comment**. **Marks** will be deducted for not writing your admission number, name and class in every source file that you are required to modify.

1. You will be given 5 minutes to:
   * 1. Download **FOP2MockPractTest.zip** from BrightSpace and unzip to your local drive. Rename the folder as **pYYYYYYMockPractTest-CLASS** where **YYYYYY** is your admin number and **CLASS** is you class eg. 1B01.

Example: p2299981PractTest-1B01

* + 1. Open the folder and check that you have the following files:

|  |  |
| --- | --- |
| **Description** | **Source Code Files Provided** |
| Client Folder | Client->SectionA->  q1.js  q2.js  q3.js  Client->SectionB->  dataVehicleInfo.js  dataVehicleType.js  q1.js |

1. **At the end of the test**, you are required to:

* **Ensure your name, class and admission number are written on all the source files modified. Marks will be deducted if you do not do so.**
* **Compress the entire “pYYYYYYMockPractTest-CLASS” folder and submit through the BrightSpace.**
* **Return the question paper back to the invigilator.**

**Mock Practical Test**

**SECTION A:**

Answer **all** questions.

1.

Christmas is around the corner, Jennifer Lim is happily creating a shopping list for her family members and relative. The following is the data store shopping list.

|  |
| --- |
| gifts = [  {  id: 0,  name: 'John',  relation: 'Brother',  gift: 'jersey',  greeting: 'Running Christmas'  },  {  id: 1,  name: 'Candice',  relation: 'Sister',  gift: 'portable fan',  greeting: 'Breeze Christmas'  },  {  id: 2,  name: 'Fat Daddy',  relation: 'Father',  gift: 'T-Shirt',  greeting: 'Fitting Christmas'  },  {  id: 3,  name: 'Love Mommy',  relation: 'Mother',  gift: 'grinder',  greeting: 'Delicious Christmas'  },  {  id: 4,  name: 'Christina',  relation: 'Aunt',  gift: 'Sling Bags',  greeting: 'Joyous Christmas'  },  ]; |

a. Write a function name myGifts() using closure.

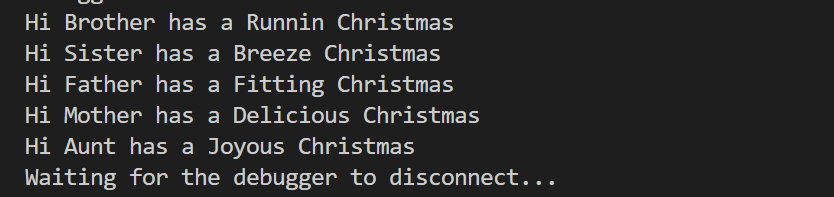
The following are the required functions in myGifts():

* add items
* delete items
* get number of gifts
* greeting message (Example: Hi <name> has a <greeting>)
* update greeting message

|  |
| --- |
| function myGifts() {  // TODO: Code here  return {  // TODO: Code here  }  }  // Display all the greeting messages.  // TODO: Code here |

b. Display all the greeting message using the myGifts() function and method greeting message in the myGifts() function.

Output:



2.

Write a higher-order function that can calculate the sum, average, median, mean and mode of a numerical array.

Variable:

operation – sum, average, median, mode

numbers – an array of data, example: num = [4, 6, 7];

Function:

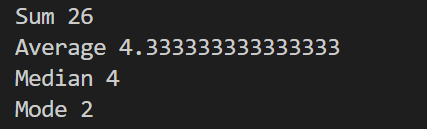
stat\_func(operation, numbers) {

}

You are given the following array to test the statvalue higher order function.

|  |  |
| --- | --- |
| operation | array |
| sum | [1, 2, 4, 3, 7, 9] |
| average | [1, 2, 4, 3, 7, 9] |
| median | [2, 4, 5, 7, 1, 8, 1] |
| mode | [2, 4, 6, 2, 2] |

Output:



3.

The set of data below is test result of a module.

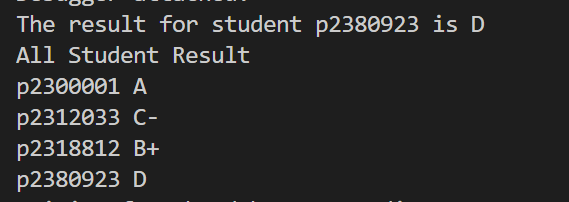
|  |
| --- |
| testresult = {  "p2300001" : "A" ,  "p2312033" : "C-" ,  "p2318812" : "B+",  "p2380923" : "D" ,  }; |

(a) Store the test result using new Map() using JavaScript. Name the new Map() studentresult.

(b) Display the result of "p2380923".

(c) List out all the student results using the studentresult Map().

Output:



**Mock Practical Test**

**SECTION B:**

Answer **all** questions.

1.

You are given a vehicle population dataset from 2005 to 2020 (dataVehicleInfo.js) and vehicle type dataset (dataVehicleType.js) in a JavaScript JSON array.

You are required to complete the following tasks.

Front-End Interface.

You are required to create a front-end interface with the following functionalities:

* Display All Vehicle Type
* Query Vehicle by Type and display the all the vehicle information throughout the years. Note: User is allowed to repeat the query. To exit the query user must enter 0 and it will return to the Main Menu.
* Query Vehicle by Year and display the total number of vehicles for the years.

You are required to use JavaScript to code the front-end interface.

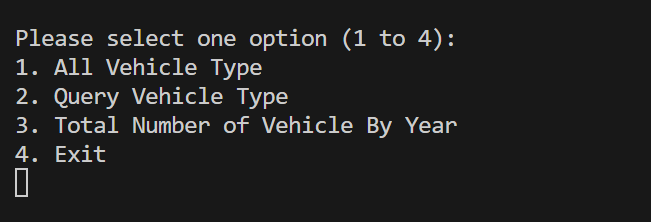
Note:

To import the dataVehicleInfo.js and dataVehicleType.js data into your code. You may use the following statement.

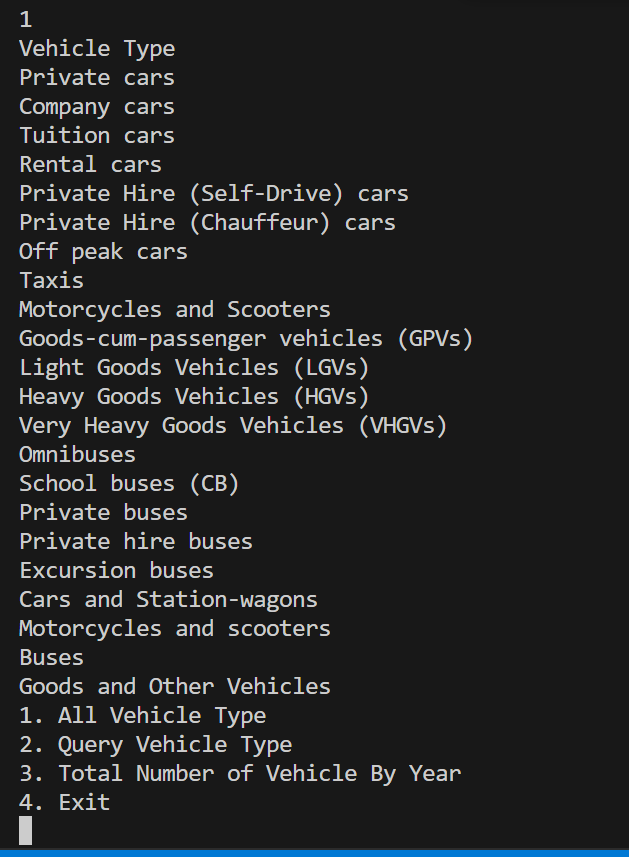
|  |
| --- |
| // All Vehicle Population data  const alldata = require('./dataVehicleInfo');  // All Vehicle Type  const allVehicleType = require('./dataVehicleType'); |

The following is the program flow:

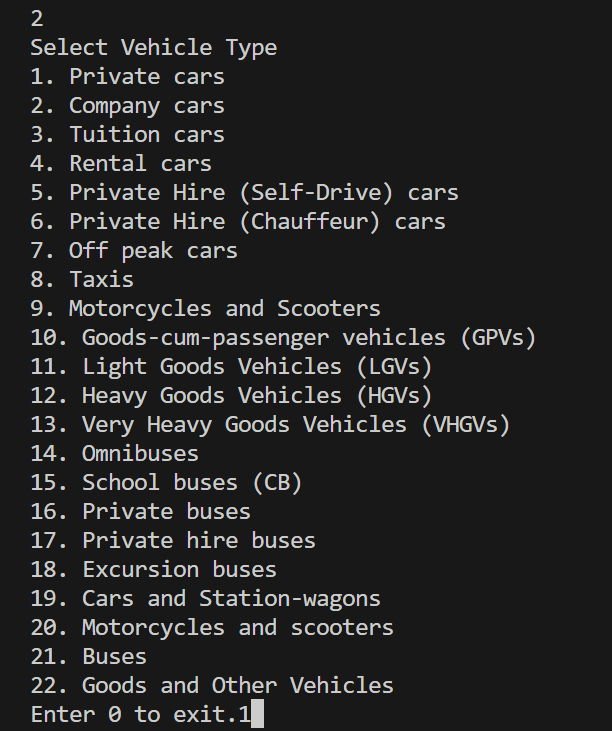
Display the Main Menu:



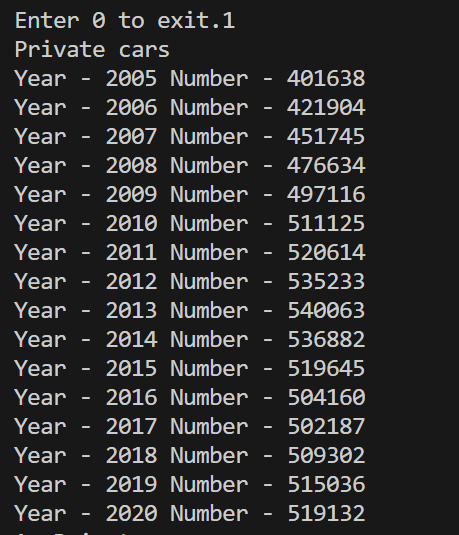
1 is Selected.



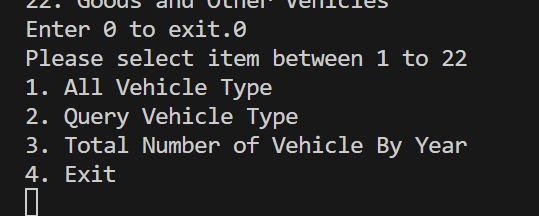
2 is Selected



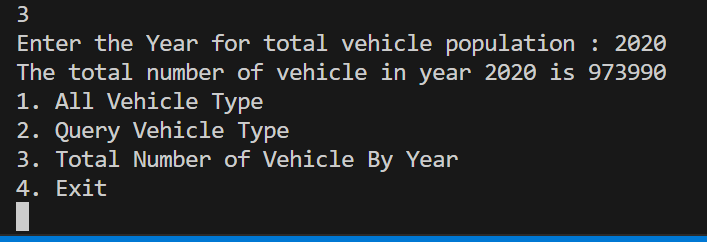
1 is Selected



0 is Selected (Return to Main Menu)



3 is Selected.



4 is Selected.

