WEB222 - Assignment 2

(1+5+6 = 12 marks)

Note: This is an individual assignment.

Due: Sunday, February 13, 2022 @11:59pm

Objective:

Practise JavaScript String, Array and customized objects.

Important note:

(1) Create a JavaScript file (name it: *a2.js*). Add the following comment declaration with your info at the top of your code: **(1 mark)**

```
/***************************************************************************
* Assignment 2
* I declare that this assignment is my own work in accordance with Seneca AcademicPolicy.
* No part of this assignment has been copied manually or electronically from any othersource
* (including web sites) or distributed to other students.
* Name:___Student ID:_______________Date: __________________
***************************************************************************/
```

(2) In *a2.js*, implement the following two tasks. Provide clear comments for each question, e.g., // Task1: a. , // Task2: a.

Task 1: (5 marks)

Create an array of objects called ***courses***. Each course in the array should have the following information:

code	name	hours	url
APC100	Applied professional communication	3	http:/www.senecacollege.ca
IPC144	Introduction to C programming	null	http:/www.senecacollege.ca/ipc
ULI101	Linux and the Internet	4	http:/www.senecacollege.ca/lin
IOS110	Windows Operating System	4	http:/www.senecacollege.ca/ios
EAC150	College English	3	null

a. Using appropriate array method, Remove the last course object from the given array ***courses*** and store the removed object to a variable.

b. On the console, show the course which was removed from the array in step a. Refer to the sample output.

Sample output:

```
The last course in the array is:
                code: EAC150,
                name: College English
```

c. Create another course object ***WEB222***, which has the same properties as the objects (in the course array). Its values are as follows:

```
code: "WEB222", name:"Web Programming Principles",
hours: 4, url: "http:/www.senecacollege.ca/web"
```

d. Add the course object ***WEB222*** to the end of the ***courses*** array.

e. Use for loop to loop through the course array in order to add the hours of the courses and log the total hours in the console. Refer to the following sample output.

Note: Do not hard code the length of the array. Your program should be easily to extend to 30 courses or more. Use the array property length.

Sample output:

```
Total hours for all courses: 15 hours.
```

Task 2: (6 marks)

a. Create a constructor function (named: ***Student***) which will be used to create objects with the following: **(1 mark)**

Properties:

`name`: a string

`dob`: built-in Date object

`sid`: a string

`program`: a string

`gpa`: a number

Methods:

Using `Student.prototype.toString()` to add the function **toString()** to the object **Student**.

Function **toString()** displays a string with student's information in the format as the following sample output.

Note: Date of Birth will be displayed in local date format (refer to built-in function: `toLocaleString()`).

Sample output:

```
Student info for name0:
                Date of Birth: 3/4/2002, 12:00:00 AM,
                Student ID: 1001,
                Program: BSD,
                GPA: 3.4
```

b. Create an empty array named **students**. **(1 mark)**
Create four student objects using the constructor function defined in step a.
One of the four student objects is “you”, with your real name, real student id, and other information as you like. Give appropriate property values for all student objects.
Do NOT directly assign the object to array elements.
Call Array function **push()** to push these four student objects into the array **students**.

c. Use the **forEach** method to iterate the array students and output the information of the student objects to the console. **(1 mark)**

d. Create a separate function (name it **highGPA()**), which doesn’t belong to the object Student. This function **(2 marks)**

** takes one parameter, which is an array of **students.**

** returns the student (object) with the highest GPA of the array **students**. If there are multiple students have the same highest GPA, returns the first occurrence.

Note: you CANNOT hard code the highest GPA or hard code the array length. Your program would be easily extend to 1000 students and find the student with the highest GPA.

Hint: use loop and array property length.

e. Display the student’s information who has the highest GPA by calling the function **highGPA(),** passing the array **students** as argument, and apply student object member function (**toString()**). **(1 mark)**

Sample output:

The student with the highest GPA is:

```
Student info for name2:
                Date of Birth: 4/4/1992, 12:00:00 AM,
                Student ID: 1001,
                Program: IFS,
                GPA: 3.8
```

Submission

Submit a2.js to Blackboard.

Late Submission

10% penalty each day for up to 5 days. After that, no submission will be accepted.