INT222 Lab 2 – Section B

Submission Deadline:

Thursday, Oct 13th, 2016 @ 11:59 PM

Assessment Weight:

5% of your final course Grade

Objective:

Work with JavaScript built-in and user-defined objects, and basic HTML.

Specification:

The lab contains three parts. You'll create two JavaScript files: one for part "A", one for part "B" and one HTML document for part "C". Complete the three parts of the lab as specified below.

Part A:

Write a JavaScript program lab02a.js to perform the following tasks. No validation is required for user input – assume that the user will enter valid information.

Open a Firefox Scratchpad. Create comment line(s) for each of the Steps in lab02a.js using block comments, indicating the start point of each Steps. e.g.

```
/*****************************
 * Step 1
 *****************************/
```

To run all JavaScript code in Scratchpad, click on the Run button. To run a part of the code, highlight the part of code and click on the Run button. You're requested to keep a Browser Console open to monitor console logs and run-time errors when running JavaScript code. Variable values will be kept in memory after a piece of code is run. So, usually, you need to initialize variables to ensure the part of code can repeatedly give the same result.

Step 1

a) Declare the following global variables without any value assigned:
e1, e2, e3, e4, e5, e6, e7, str

b) Run the code the in Firefox Scratchpad to test if you code has any errors or incorrect output. Fix the errors before going to the next step.

Step 2

a) Create a function named **capFirstLetter** using the function declaration syntax. The function receives a single parameter of String type. Update / change the first letter of the string to upper case and other letters to lower case. The function returns the updated String.

b) Write code to prompt the user to enter first name, and use your first name as default value. Accept/store the entered name in **e1**.

c) Update / change the first letter in **e1** to upper case and other letters to lower case by invoking the function.

d) Repeat step 1.b. (run the code and check for incorrect output & errors)

Hint: use the property and methods of String object – length, substr(from, length), substring(from, to) , toUpperCase() and/or toLowerCase().

Step 3

a) Create a function named **getAge** using the function expression syntax. This function receives one parameter of integer (number type), which is the year of a person's birth day. The function returns the age which is calculated based on the year entered.

b) Prompt the user to enter the year of the user's birth day – accept the number in **e2**. For the default value, use the year when you were born.

c) Calculate the age by calling the getAge() function and assign the number of age back to **e2**.

d) Repeat step 1.b. (run the code and check for incorrect output & errors)

Hint: for getting the number of the current year, you must use the code: **(new Date()).getFullYear()**

Step 4

a) Prompt the user to enter the college name the user is attending and assign the input to **e3**. For its default value, use **Seneca College.**

b) Change the first letter of each word of the string in variable e3 to upper case and other letters to lower case.

c) Repeat step 1.b. (run the code and check for incorrect output & errors)

Hint: use the split() method of String and the capFirstLetter() function you created.

Step 5

a) Prompt the user to enter 5 favorite sports (in lower case separated by comma) - accept the string in **e4**. Use **hockey,football,basketball,tennis,golf** as default value for the prompt.

b) If the string in **e4** contains "**football**", replace it with the string "**soccer**"

c) Split the sports in **e4** into an array and store the array back in **e4**.

d) Prompt the user to enter an extra favorite sport with the default value "**formula 1**" – accept it in **e5**. Then add the sport (**e5**) at the end of the course array (**e4**).

e) Repeat step 1.b. (run the code and check for incorrect output & errors)

Hint: use the split() and replace() method of String; use the push() method of Array.

Step 6

a) For the courses stored in **e4**, do the following operations.

- Update / change each sport string in the array to upper case.
- Sort the courses in the array in alphabetical order.

b) Repeat step 1.b. (run the code and check for incorrect output & errors)

Hint: use sort() method of Array object.

Step 7

a) Create a function named **getDateString**(). This function receives one parameter of Date type and returns date string with the format of **yyyy-mm-dd**. e.g. 2016-09-20.

Note: if the number of the month (mm) or date (dd) is less than 10, a '0' is needed before the number.

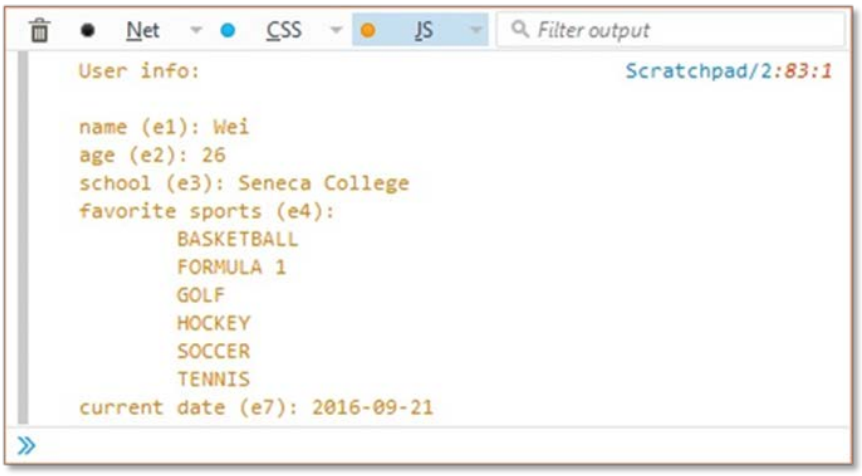
b) Create a date object with current date and time, and store it to **e6**.

c) Get current date string with the format of yyyy-mm-dd by calling the **getDateString**() function and passing **e6** as parameter. Store the date string in **e7**.

Step 8

a) Concatenate all the variables **e1, e2, e3, e4** and **e7** with appropriate text in variable **str**.

b) Use one statement **console.log(str);** to get the following output:



c) Save your file as lab02a.js.

Hint: use '\n' and '\t' to create multiple lines and indents in browser/web console.

Part B:

Locate the file **lab02b.js** (available in the .zip file next to this document), which contains some given code, including an array (named **courses**) of course objects and a prototype object (named **student**) for creating student objects. Do not change the given code. Write your code beneath the given code and complete the following tasks:

Task 1

a) Remove the last course object form the given array courses and store the removed object to a variable.

b) Output a message to browser console to show which course was removed from the array. Please see the screenshot of outputs on the browse console below.

c) Create 4 course objects that have the same properties as the course objects in the array have. Store the 4 course objects in the variables **ibc233**, **oop244**, **int222** and **dbs201** and give appropriate values for their proprieties.

d) Add these course objects in the array courses.

e) Use for loop to loop through the course array and output the information of the course objects in the array to browser console. Please refer the screenshot below.

Task 2

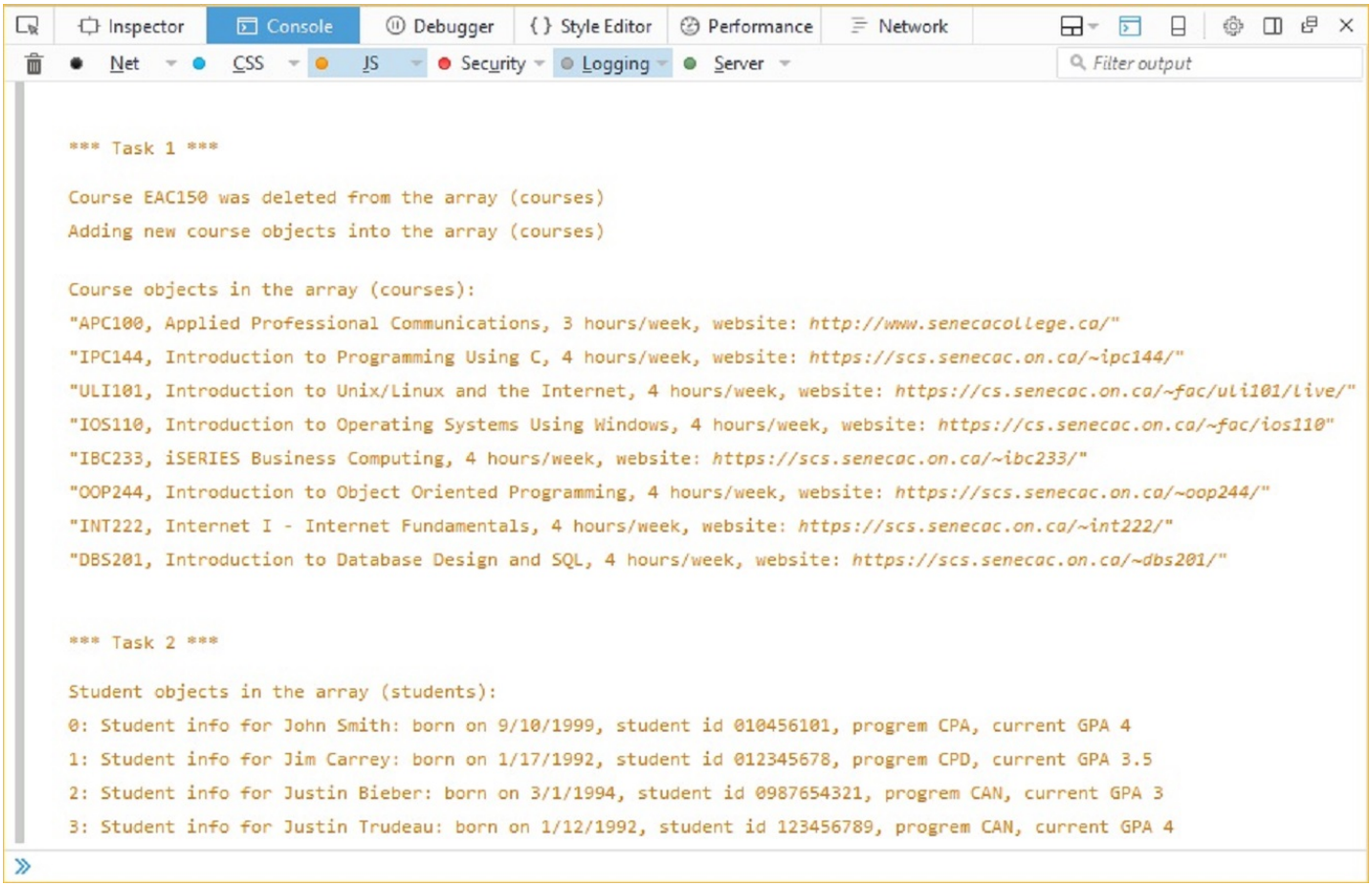
a) Create 4 student objects based on the given prototype student. Give appropriate property values for all student objects.

b) Create an array named students and add all the student objects into the array.

c) Loop through the array of students and output the information of each student object to the browser console. Please refer the screen below.

After you complete the both tasks (Task 1 and Task 2), save your JavaScript code to file lab02b.js.

Part B – Output Sample (See lab02b-output.jpg for a larger version)



Part C:

Locate the file **lab02c.html** (available in the .zip file next to this document), which contains a basic HTML page without any content. Do not change the given code. Write your new code within the **<body></body>** tags and complete the following tasks (NOTE: Complete the tasks in order to ensure that your page looks like the expected result (see **Part C Output Sample** below)):

Task 1

a) Change the page title: **???'s HTML Playground** to use your name in place of ???, ie: **Pat's HTML Playground**

b) Show a **level-1 header** at the top of the page with the text: **???'s HTML Playground** (where ??? is your name)

c) Show a paragraph with the text: "**Welcome to ???'s HTML Playground! Here, we will show some examples of common HTML Elements:**"

Task 2

a) Create 4 **page divisions**, with the following id attributes: "**partA**", "**partB**", "**partC**", "**partD**"

b) In the "**partA**" division, create the following elements:

- A **level-2 header** with the text: "**<blockquote>...</blockquote>"** (Hint: You will have to make use of the < and > HTML entities to show the characters: < and >
- A **paragraph** element, containing the text: "**Here is a paragraph tag, sitting above our blockquote element**"
- A **blockquote** element, containing the text: "**I'm in a blockquote! – ???**", where ??? is a short quote
- A **paragraph** element, containing the text: "**Here is a paragraph tag, sitting below our blockquote element**"
- A **horizontal rule** tag helping to show the end of this segment

c) In the "**partB**" division, create the following elements:

- A **level-2 header** with the text: "**<pre>...</pre>"** (Hint: You will have to make use of the < and > HTML entities to show the characters: **<** and **>**
- A **paragraph** element, containing the text: "**Here is a paragraph tag, sitting above our pre element**"
- A **pre** element , containing the text (and preserving the whitespace):

```
var sayHi = function(message){
        console.log(message);
}

sayHi("Hello from pre!");
```

- A **paragraph** element, containing the text: "**Here is a paragraph tag, sitting below our pre element**"
- A **horizontal rule** tag helping to show the end of this segment

d) In the "**partC**" division, create the following elements:

- A **level-2 header** with the text: "**Presentation Tags: <b>, , <i>, <u>"** (Hint: You will have to make use of the < and > HTML entities to show the characters: **<** and **>**
- A **paragraph** element, making use of the **<b>, , <i>,** & **<u>** tags, resulting in the text: **"**Here is a paragraph with **bold** items, *emphasized* items, *italic* items, <u>underlined</u> items."
- A **paragraph** element, making use of the **<b>, , <i>,** & **<u>** tags all at once, resulting in the text: "Here's another with ***<u>all 4 in one</u>***"
- A **horizontal rule** tag helping to show the end of this segment

e) In the "**partD**" division, create the following elements:

- A **level-2 header** with the text: "**Lists: <ol>, <ul>, <dl>"** (Hint: You will have to make use of the < and > HTML entities to show the characters: **<** and **>**

- A **Ordered List** element containing two **List Items** listing two separate sports, ie: "**Baseball**" and "**Football**"
    - Nested within the first Ordered **List Item**, add an **Unordered List** element containing two **list items** showing teams that play the first sport, ie "**Blue Jays**" and "**Yankees**"
    - Nested within the second Ordered **List Item**, add an **Unordered List** element containing two **list items** showing teams that play the second sport, ie "**Argonauts**" and **"Tiger-Cats"**
- A **Definition List** element, containing two **definitions**:
    - One definition with the title "**HTML**" and description "**HTML, which stands for HyperText Markup Language, is the most basic building block of a webpage and used for creating and visually representing a webpage.**". The description must also contain a link to: https://developer.mozilla.org/en-US/docs/Web/HTML using the text "**source**". Make sure this link opens in a new tab/window.
    - A Second definition with the title "**CSS**" and description " **Cascading Style Sheets (CSS) are a stylesheet language used to describe the presentation of a document written in HTML or XML**". The description must also contain a link to: https://developer.mozilla.org/en-US/docs/Web/CSS using the text "**source**". Make sure this link opens in a new tab/window.

Task 3

a) Validate your HTML using: https://validator.w3.org/#validate_by_input and fix any errors – your submitted HTML file must not contain any errors.

Part C – Output Sample (See lab02c-output.jpg for a larger version)

Pat's HTML Playground

Welcome to Pat's HTML Playground! Here, we will show some examples of common HTML Elements:

<blockquote>...</blockquote>

Here is a paragraph tag, sitting above our blockquote element

> I'm in a blockquote! - "Time is an illusion"

Here is a paragraph tag, sitting below our blockquote element

---

<pre>...</pre>

Here is a paragraph tag, sitting above our pre element

```
var sayHi = function(message){
        console.log(message);
}

sayHi("Hello from pre!");
```

Here is a paragraph tag, sitting below our pre element

Presentation Tags: <b>, , <i>, <u>

Here is a paragraph with **bold** items, *emphasized* items, *italic* items, underlined items.

Here's another with ***all 4 in one***

Lists: <ol>, <ul>, <dl>

1. Baseball
   - Blue Jays
   - Yankees
2. Football
   - Argonauts
   - Tiger-Cats

HTML
: HTML, which stands for HyperText Markup Language, is the most basic building block of a webpage and used for creating and visually representing a webpage. - source

CSS
: Cascading Style Sheets (CSS) are a stylesheet language used to describe the presentation of a document written in HTML or XML - source

Lab Submission:

- Add the following declaration at the top of your lab02a.js, and your lab02b.js files:

```
/*********************************************************************************
* INT222 – Lab 02
* I declare that this lab is my own work in accordance with Seneca  Academic Policy.  No part of this
* lab has been copied manually or electronically from any other source (including web sites) or
* distributed to other students.
*
* Name: _____ Student ID: _____ Date: _____
*
*********************************************************************************/
```

- Complete the student declaration provided at the top of lab02c.html (same as above)

- Compact your files **lab02a.js, lab02b.js** and **lab02c.html** into a zip file named **lab02.zip**.
- Submit your **lab02.zip** file to Blackboard (My.Seneca) under **Labs** > **Lab 2**

Important Note:

- Your HTML Must pass the validation here: https://validator.w3.org/#validate_by_input without any errors
- Your JavaScript files must not show any errors when run in the browser.
- **NO LATE SUBMISSIONS** for labs. Late Lab submissions will not be accepted and will receive a **grade of zero (0)**.
- After the end (11:59PM) of the due date, the lab submission link on the Blackboard will no longer be available.