Assignment 1 - Ver1.0

Released on: Friday, May 29th, 2015
Due on: Saturday, June 20th, 2015 23:59
Value: 6% of final mark
Late penalty: 10% for the 1st delay day; 5% for each day thereafter

Note: This document may be updated from time to time. Please check the version number when reading the requirements. You should be responsible to maintain your assignment to meet the latest requirements.

OVERVIEW

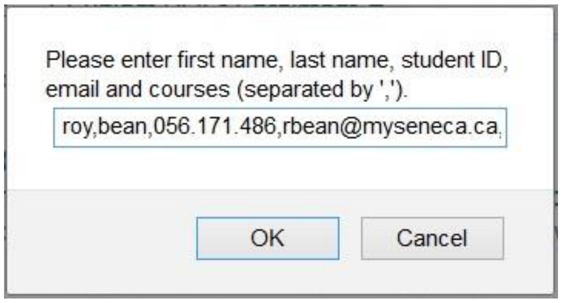
In this assignment, you will create a JavaScript program named **as1.js** which let user input student registration information and query student list for each course. This program is supposed to run within Firefox Scratchpad. You are required to upload the completed file to your account on zenit server for evaluation.

OBJECTIVES

- Practice basic programming in JavaScript.
- Understand and apply the concept of function and object in JavaScript.

SPECIFICATIONS

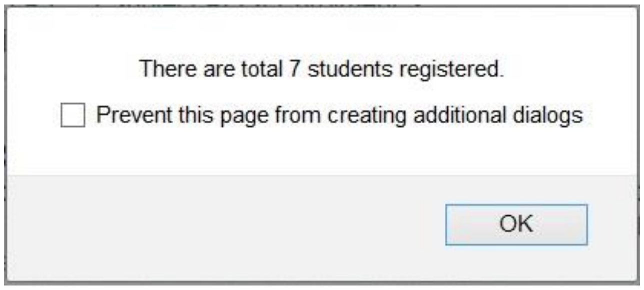
1. The JavaScript program starts by continually prompting to ask user to enter student registration information until the user clicks on the "Cancel" button or enter a blank text. The dialog box is showed as following:


Please enter first name, last name, student ID, email and courses (separated by ',').

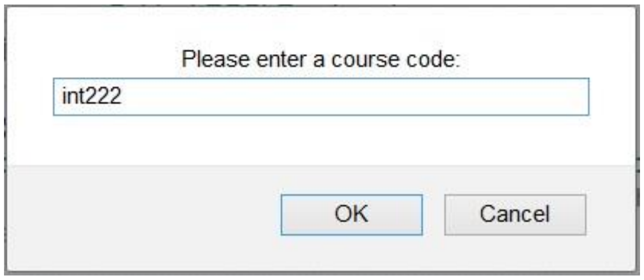
roy,bean,056.171.486,rbean@myseneca.ca,

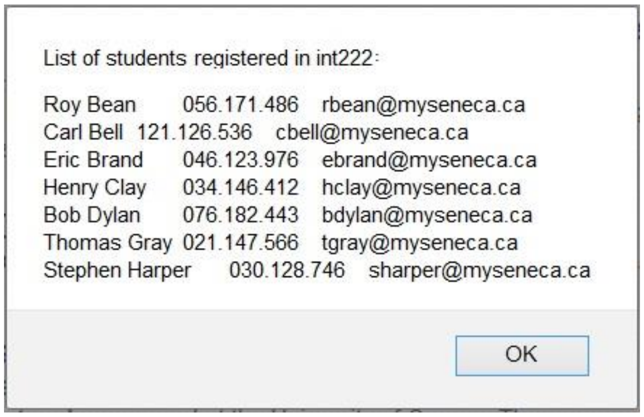
2. The student registration information includes **first name**, **last name**, **student number**, **email address** and at least one **course code** registered in the current semester. These data fields have to follow the sequence showed above and separated by comma (','). Here is a sample of a valid input:
   - ✓ roy,bean,056.171.486,rbean@myseneca.ca,eac150,DBS201,int222

3. Your JavaScript program should parse the user's inputs. For each input, you should create a JavaScript object of Student to hold the data parsed from user's input. Make sure that each object should have a property named "**courses**" which should have an string array of course codes as value, e.g. ["EAC150", "DBS201", "INT222"] for the sample input above.
4. Upon the user clicks on the "Cancel" button or enter a blank text, a dialog will pop up to show the number of students registered, for example:

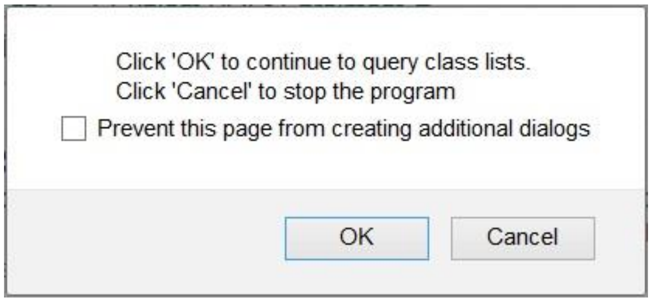

There are total 7 students registered.

5. Then the program will enter a state to continually allow user to query student list by typing course code. The process is as following:


Please enter a course code:
int222



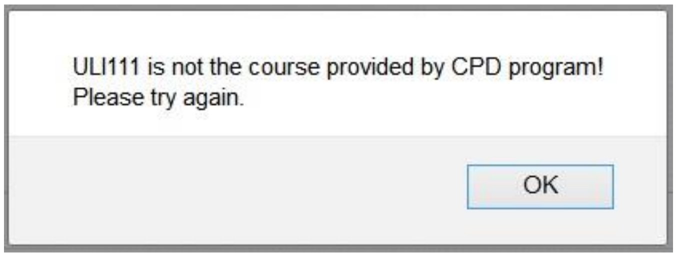
```
List of students registered in int222:

Roy Bean      056.171.486   rbean@myseneca.ca
Carl Bell  121.126.536   cbell@myseneca.ca
Eric Brand    046.123.976   ebrand@myseneca.ca
Henry Clay    034.146.412   hclay@myseneca.ca
Bob Dylan     076.182.443   bdylan@myseneca.ca
Thomas Gray 021.147.566   tgray@myseneca.ca
Stephen Harper      030.128.746   sharper@myseneca.ca
```


Click 'OK' to continue to query class lists.
Click 'Cancel' to stop the program

REQUIREMENTS

- You are requested to create the following functions to complete the assignment:
    1. formatingName(name) – the function accepts the parameter "name" which can be first name or last name in a single word. The function will return the name with requested format: the first letter capitalized and the rest in lower case.
    2. validateStudentID(sid) – this function receives the parameter of student id and return true or false for valid or invalid. Student ID – the third field of the input string – must be taken the format of xxx.xxx.xxx, where x means digit (0-9). You are required to use RegExp method(s) to implement this function.
    3. validateCourses(courses) – this function receives a string array of course codes which are to be registered for a student. The function returns an empty string indicating all course codes in the array are valid; otherwise the function returns the first invalid course code in the array. All course codes must be selected from the course codes provided for the first and second semesters of the CPA/CPD program. They are: APC100, IPC144, ULI101, IOS110, EAC150, IBC233, OOP244, DBS201 and INT222.
- The Student object must provide a function member named **hasCourse**. This function takes a single course code as parameter. The function will return true if the student has registered this course (code) and returns false if not.
- A student must have at least one course for registering. Otherwise the input data will be invalid. The max number of courses for registering is six. If the number is greater than 6, the input data should be valid, but the course code(s) beyond 6 will not be considered.
- Your program should allow user to input letters in upper or lower cases. User may put space(s) around the divider – comma ','.
- The data stored in a Student object must be in the required formats:

    Letters in a course code should be in upper case.
    Letters in an email should be in lower case.
    Each first and last name need to be processed by using **formattingName** function.
- Your program must show appropriate message to user for each invalid input and allow user to continue to re-input data. e.g.


ULI111 is not the course provided by CPD program!
Please try again.

- **Advanced:** you may create a **Person** object with first name, last name and email as properties. Then use the Person object as prototype to create your **Student** objects. In addition to the properties inherited from **person** object, Student objects should have two more properties: **sid** (for student ID) and **courses**.

- No error or exception is allowed when your program is running in Scratchpad. Watch out Brower Console all the time for errors or exceptions. Runtime errors or exceptions will result in your assignment incomplete.
- You are suggested to use JSLint to validate you JS code.

TEST DATA (VALID AND INVALID):

```
roy,bean,056.171.486,rbean@myseneca.ca,int222
carl,bell,121.126.536,cbell@myseneca.ca,dbs201,int222
eric,brand,046.123.976,ebrand@myseneca.ca,oop244,dbs201,int222
henry, clay, 034.146.412, hclay@myseneca.ca , ibc233 , oop244 , dbs201 , int222
bob,dylan,076.182.443,bdylan@myseneca.ca, ibc233, oop244, dbs201, int222, eac150
thomas,gray,021.147.566,tgray@myseneca.ca,ios110,ibc233,oop244,dbs201,int222,eac150
stephen,harper,030.128.746,sharper@myseneca.ca,uli101,ios110,ibc233,oop244,dbs201,int222,eac150

joe,hill,O3O.152.471,jhill@myseneca.ca,int222,ibc233,ibc233,oop244,dbs201
steve,jobs,010.125.236,sjobs@myseneca.ca,iOS201
jhon,smith,012.023.165,jsmith@myseneca.ca
```

SUBMISSION

In the **as1.js** file, add the following declaration at the top of your code:

```
/*********************************************************************************
 *                      INT222 - Assignment #1
 * I declare that this assignment is my own work in accordance with Seneca
 * Academic Policy.  No part of this assignment has been copied manually or
 * electronically from any other source (including web sites) or distributed to
 * other students.
 *
 * Name: ______________________ Student ID: ______________ Date: ________________
 *
 ********************************************************************************/
```

Upload your **as1.js** under the **public_html/assign1** directory of your **zenit** account before the due date.