**Student ID:** 985892

**Full Names:** Dulamkhand Batjargal

Web Application Programming

(CS472)

(September 2017)

Instructor: Obinna A. Kalu

W2D6 – Exam 2

1. The exam duration is 2 hours.
2. The exam is a computer-based exam.
3. You are expected to use a CS lab or your own computer to answer both the Coding questions and the theory/non-coding/knowledge-based questions. You may use the Internet and/or the lecture slides for reference purposes to lookup APIs or code syntax.
4. Make sure to switch-off your cell-phones or simply turn the ringer off.
5. You may use blank sheet(s) of paper for your scratch work, if needed.
6. Exams are copyrighted materials and must not be copied, reproduced or redistributed.
7. All answers to the theory/non-coding/knowledge-based questions should be typed-in, on this document, following the questions.
8. All answers to the Coding questions may be typed-in as source code files, using a Code Editor or IDE. But be sure to copy your finished code for each coding question from your Code Editor and paste it to this document as your answer.
9. Finally, compress/zip your entire code folder into one zip file and upload/submit it to Sakai, along with your typed/pasted answers in this document.

--------------------------------------------------------------------------------------------------------------------

(CS472 - WAP)

(September 2017)

W2D6 – Examination 2

**Part I – Science of Consciousness (SCI):** (5 points)

Given in the table below, are principles from the Science of Consciousness (SCI), related topics in Web Application Programming (WAP) and Connection sentence(s). For each section, fill-in the blank cell(s), by either providing the missing principle from the Science of Consciousness or a Web Programming topic or a connection sentence(s).

**Note**: To get the full credit, make sure your connection sentence(s) provide(s) a reasonable, clear link/relation between the stated SCI principle and the WAP topic.

|  |  |  |
| --- | --- | --- |
| **SCI principle** | **Topic from Web Application Programming (WAP)** | **Connection sentence(s)** |
|  | Inheritance in JavaScript |  |
| Seek the highest first |  | Mastery of fundamental web programming concepts gives us an edge. Likewise, when we seek the highest first we achieve more out of life. Through regular practice of Transcendental Meditation we gain maximum advantage. |
|  | jQuery |  |

**Part II – Theory (Short answers, True/False, Multiple-choice questions):** (35 points)

1. (8 points) Answer the following questions with True or False. For each answer, give a rationale (i.e. If True state how, if False state why. No rationale, earns you just half of the points if your True/False answer is correct, and zero point if your True/False answer is incorrect).
   1. (2 points) In JavaScript, when a function is defined inside another function, the outer function has access to the inner function’s variables.

False - Inner functions have access to the scope of outer function, but not with other way.

* 1. (2 points) By using the jQuery selector, $(“<div>”), we can select all the div tags present on our webpage.

True – The correct selection is $(“div”).

* 1. (2 points) When using the Web Storage API, the localStorage containing data stored by the website, <http://www.mum.edu/> CANNOT be accessed by your webpage at <http://mumstudents.org/>.

True -

* 1. (2 points) In Javascript, when a variable is declared, (e.g – var dateOfBirth; ) and no value is assigned to it, the default value it gets is, null.

False – The default value is undefined.

1. (20 points) Give short answers (and examples where required) to the following questions.
   1. (2 points) Does JavaScript support function overloading? Explain your answer + give an example using code snippet.

Javascript does not support function overloading. When there is two or more functions with same names JS consider the last defined function and overwrite the previous ones. Because parameters for the function are optional. For instance:

TODO

* 1. (4 points) Explain the difference between Obstrusive versus Unobstrusive Javascript. Give an example of each, to illustrate your answer.

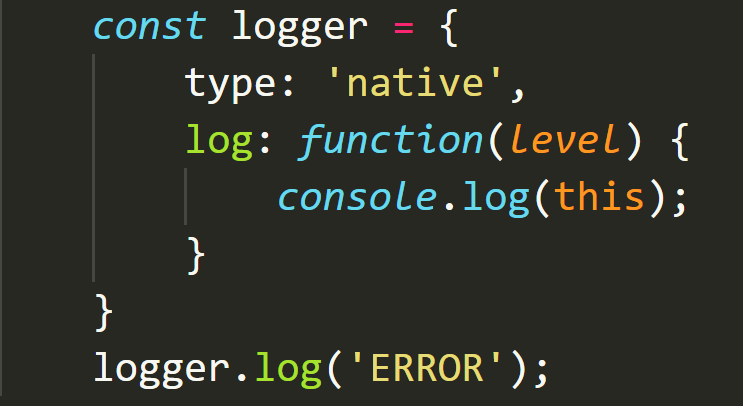
Unobstrusive way is good, it separates HTML, CSS and JavaScript.

Obstrusive is avoidable as javascript tied with HTML.

<p onclick="alert('obstrusive')">Obtrusive Example</div>

* 1. (6 points) In terms of Javascript programmming:
     1. What is a Closure? Give an example in code snippet.
     2. What is an IIFE? Give an example in code snippet.
     3. What is the Module Pattern? Give an example in code snippet.
  2. (2 points) In Web Application programming, what do we mean by **AJAX** (You may give an example to illustrate your answer).
  3. (4 points) With reference to Inheritance in JavaScript, what is the difference & the relationship between the .\_\_proto\_\_ property and the .prototype property? (You may give an example to illustrate your answer).
  4. (2 points) With regards to JavaScript’s support for Functional programming, what is a First class function? Give an example.

1. (7 points) The following questions involve multiple choices; choose the correct option by putting a green highlight over, either Option A or Option B or Option C.
   1. (2 points) Consider the code snippet given below:



When the statement, *logger.log(‘ERROR’)* is executed, what object will the ‘this’ reference point to?

**Option A**.

window object

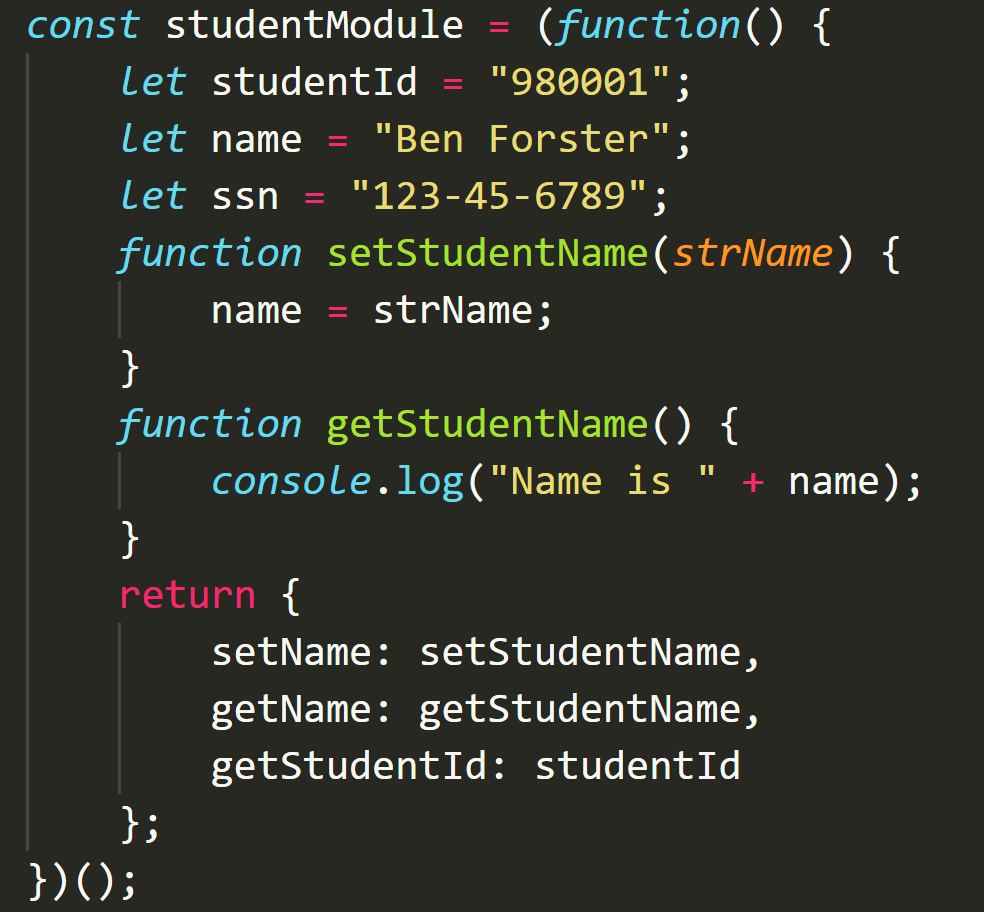
**Option B**.

logger object

**Option C**.

ERROR object

* 1. (3 points) Based on the code snippet given below, which is the incorrect statement:



**Option A**.

getStudentName is a public function

**Option B**.

studentId is a private variable

**Option C**.

setName is a public function

* 1. (2 points) Consider the HTML markup code-snippet for the submit button element given below.

<button id="btnSubmit" type="submit">Submit</button>

Which option is NOT the recommended best practice for adding its click event handler?

**Option A**.

<button id="btnSubmit" type="submit" onclick=”addStudent();”>Submit</button>

**Option B**.

document.querySelector(“#btnSubmit”).addEventListener(“click”, function(event) {…})

**Option C**.

$(“#btnSubmit).on(“click”, function(event) {…})

**Part III – Skill (JavaScript/Web Coding):** (60 points)

1. (5 points) Write a JavaScript function named, *calculateAverage*, that takes any variable amount of numbers, as input, and returns the average. E.g. calculateAverage(1,2,3) should return 3, calculateAverage(1,2,3,4) should return 2.5 and calculateAverage(1,2,3,4,5,16.6) should return 5.26666, and so on.
2. (5 points) Using JavaScript functional programming style, write a function named, multiplesCount, that takes an array of numbers, nums and an integer base, m, and returns how many of those numbers are multiples of m. e.g. multiplesCount([1,2,3,4,5,6], 3) should return 2, multiplesCount([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15], 5) should return 3 and so on. **[Note: Make sure your solution is a functional programming approach; No use of ‘for…loop’ or ‘if’ statements allowed]**
3. (10 points) **JavaScript Inheritance – redHondaAccord IS-A Car**:
   1. (5points) Define a base object named, **Car**, as object literal, with the following properties – **make** (set its initial value as ‘default’), **model** (set its initial value as ‘default’), **color** (set its initial value as ‘default’), and include the following two methods:
      1. drive - takes a parameter named, speed, and prints-out the following message to the console – “The [color] [make] [model] is driving at [speed] mph.
      2. stop – prints-out the following message to the console – “The [color] [make] [model] is stopping.

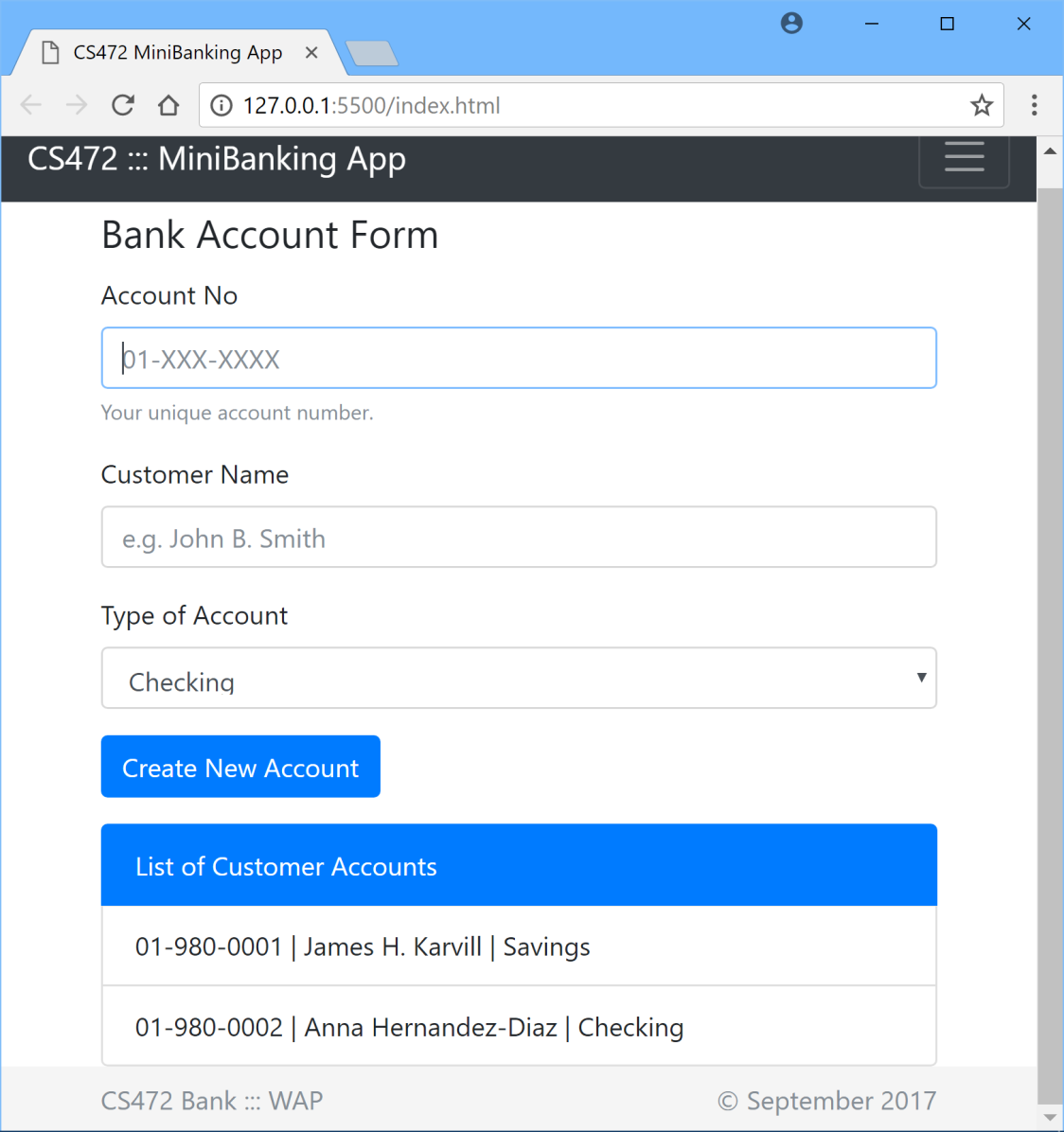
Using inheritance, create a derived object named, redHondaAccord, which inherits from the Car object.

Assign the values ‘red’, ‘Honda’ and ‘Accord’ to the respective properties of color, make, model for the redHondaAccord object.

Invoke the drive(…) method on the redHondaAccord object, passing it a **speed** value of 200.

* 1. (5points) Re-implement the above described inheritance structure, but this time, using a Constructor function. i.e. define a constructor function named Car and then add to it, a drive(…) method and a stop() method. Next, based on your constructor function, create a derived object named redHondaAccord, with red, Honda and Accord for color, make and model respectively. Finally, invoke its drive method passing in the value of speed, 200.

1. (40 points) Using HTML, CSS and JavaScript, implement a mini-BankAccount single-page web application, as shown in the UI screenshot below, with the following features and functionalities:



1. Code the User interface of the webapp using standard, semantically correct HTML5 markup, including all the form fields as shown above.
2. Apply styling using Bootstrap or you may apply your own custom CSS styling to produce the same/similar look and layout.
3. Account Number and Customer Name are mandatory to create a new account.
4. Add validation using appropriate regular expression to ensure that any Account Number entered is in the specified format, 01-XXX-XXXX, as shown on the UI above.
5. The user can create 2 types of Bank Accounts – Checking and Savings.
6. Use JQuery and AJAX to load the List of existing Customer Accounts, by making an AJAX HttpGET request to a JSON-formatted data file located at the url – <http://yourApplicationHostAddress/data/customerData.json>.
7. Implement the ‘Create New Account’ functionality to add a new account data to the List on the webpage.
8. Add the Customer Accounts data to localStorage, such that when the browser is closed and re-opened, the webapp still displays all customer Account data, including any newly added accounts.

*(Please note: Points will be awarded based on your adherence to Web programming recommended best practices such as use of unobstrusive CSS, unobstrusive JS etc.*

***Also, make sure all your coding for this question is done in One webapp root folder named P3Q4, contained inside your Exam2 folder****)*

**//-- The End --//**