**Exam 2 Practice/Review Questions**

1. 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. In JavaScript, when a function is defined inside another function, the outer function has access to the inner function’s variables. True or **False**?
   2. By using the jQuery selector, $(“<div>”), we can select all the div tags present on our webpage. **True** or False?
   3. 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** or False?
   4. 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. True or **False**?
2. Give short answers (and examples where required) to the following questions:
   1. Does JavaScript support function overloading? Explain your answer by giving an example using code.

**It’s not supported. For example**

**Function log(parameter) {**

**Console.log(parameter);**

**}**

**Function log(data, parameter) {**

**Console.log(data, parameter);**

**}**

**These are doesn’t work because functions are objects.**

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

**Obstrusive event handler is that attaches event to element in the HTML. Unobstrusive is attaches event to element in the JS.**

**Function sayHi(){**

**Console.log(“Hi”);**

**}**

**Obstrusive : <button onclick=”sayHi();”>Say Hi</button>**

**Unobstrusive: <button id=”sayHi”>Say Hi</button>**

**Var sayHi = document.getElementById(“sayHi”);**

**sayHi.onclick = sayHi;**

* 1. In terms of Javascript programmming:
     1. What is a Closure? Give an example in code.

**Closure is a first class function that binds to free variables that are defined in its execution environment.**

* + 1. What is an IIFE? Give an example in code.

**IIFE means Immediately Invoked Function Expression. Furthermore, It declares function and immediately calls a anonymous function**

**(function(parameters) {**

**Console.log(paramaters);  
})(parameters);**

* + 1. What is the Module Pattern? Give an example in code.

**Module Pattern uses IIFE for the purpose that encapsulate methods and properties.**

**Var counter = (function() {**

**Var privateCounter = 0;//private property**

**Function changeBy(val) { //private method  
 privateCounter +=val;**

**}**

**Return {**

**Increment: function() {changeBy(1);}**

**Decrement: function() {changeBy(-1);}**

**Value:function() {return privateCounter;},**

**publicProperty: 5// example of public property**

**}**

**}();**

* 1. In Web Application programming, what do we mean by **AJAX** (You may give an example to illustrate your answer).

**AJAX means Asynchronous Javascript and XML. With ajax, web application can send and retrieve data from a server asynchronously which means in the background.**

* 1. With reference to Inheritance in JavaScript, what is the difference and the relationship between the .*\_\_proto\_\_* property and the .*prototype* property? (You may give an example to illustrate your answer).

**\_\_proto\_\_ is the actual object that is used in the lookup chain to resolve methods, etc. prototype is the object that is used to build \_\_proto\_\_ when you create an object with new**

**( new Foo ).\_\_proto\_\_ === Foo.prototype;**

**( new Foo ).prototype === undefined;**

* 1. With regards to JavaScript’s support for Functional programming, what is a First class function? Give an example.

**First class function means when function in that language treated like any other variable.For example, a function can passed as parameter to other functions, can returned by another function can be assigned as variable.**

* + 1. **Const foo =function(){}**
    2. **Function a(paramater){**

**console.log(paramater);**

**}**

**Function b (func, parameter) {**

**return func(parameter);**

**}**

**b(a, ‘haha’);**

* + 1. **Function a() {**

**Return function() {return ‘haha’;}**

**}**

* 1. Write a code example of a Function declaration

**Function declare() {**

**return ‘asdasd’;**

**}**

* 1. Write a code example of a Function Expression

**Const expression = function() {**

**Return ‘asd’;**

**}**

* 1. Write a code example of an Anonymous Function

**Function a() {**

**Return function() {return ‘haha’;}**

**}**

* 1. Write a code example of an Arrow Function

**Const arrow = (parameter) => {**

**Return parameter;**

**}**

* 1. In JavaScript programming, what is referred to as, "Hoisting"? Give an example.

**Hoisting is JS engine reserve special memory for functions(as whole) and variable names. Variables initialy set to undefined**

* 1. With reference to the JavaScript language, what is a Lexical Environment? Execution Context?

**The lexical environment is where your JS codes sitting physically. Every lexical environment has own execution context which that code running.**

* 1. With reference to JavaScript language versions, ES6 and above, how many possible Scopes are there?
     1. **Global**
     2. **Block/which used by let and const/**
  2. Assume that a JavaScript source code file named, app.js, is added to a web page file named, index.html, using the <script> tag. If the first and only line of code is app.js is:

console.log(this);

When the index.html page is opened in a web browser, what object will be printed to the console?

**It will not work because javascript is fully event driven language. You need to fire event to execute app.js.**

* 1. In client-side Web Application programming, what is referred to as "Event Bubbling"? Give an example.

**Event bubbling means the event captured and handled by the innermost element to the outer elements.**

**<div>**

**<p>text</p>**

**</div>**

**If we set onclick event on each element and click on element It will execute p element’s onclick event and then div’s onclick event.**

* 1. How do we prevent a click event on a hyperlink element from performing its default behavior of navigating to the url referenced by its href attribute?

**<a href=**[**http://www.test.com**](http://www.test.com) **onclick=”event.preventDefault();” >Anchor tag </a>**

* 1. The JavaScript Function object has 3 important/essential methods, namely,

Function.prototype.call(...)

Function.prototype.apply(...)

Function.prototype.bind(...)

Briefly explain the purpose of each of these methods, by giving small code examples to support your answer.

**Bind-> creates copy of function with args already set**

**Call-> calls function with give this value and arguments provided individually**

**Apply-> calls function with give this value and arguments provided as and array**

**var me = {**

**first: 'Asaad',**

**last: 'Saad',**

**getFullName: function() { return this.first + ' ' + this.last; }**

**};**

**var log = function(height, weight) { // ‘this’ refers to the invoker**

**console.log(this.getFullName() + height + ' ' + weight);**

**};**

**log.call(me, '180cm', '70kg'); // Asaad Saad 180cm 70kg**

**log.apply(me, ['180cm', '70kg']); // Asaad Saad 180cm 70kg**

**var logMe = log.bind(me);**

**logMe('180cm'); // Asaad Saad 180cm undefined**

**var logMeWeight = log.bind(me, '180cm');**

**logMeWeight('70kg'); // Asaad Saad 180cm 70kg**

* 1. Assume you want to use the Location API in your web application. What is the name of the *object.property* to use to check if the user's browser has support for this API?

**navigator.geolocation.getCurrentPosition()**

1. JavaScript OOP:

Create a new webapp. Include an index.html page and load a JavaScript file named, app.js. In app.js, **define a Javascript object** **literal** named, person with the following specification:

person data fields (properties):

- name (default value: "")

- dob (default value: today's date)

- age (default value: 0)

person methods:

- getName : should return the name value of the person object

- setName : should assign name with value of an input param, named, newName.

Next, create an object named, John, who IS-A person, with the following data:

name is John,

dob is 1998-11-09

age is 21

Next, print to the console, John's information as follows:

"John born on 1998-11-9 is 21 years"

Given that an employee IS-A person. Create an object named, employee. Define the following new properties for employee object:

- salary: 0.0

- hireDate: today's date

Then, add a new method to the employee object named, doJob, which takes a param named, jobTitle. And when doJob is called, the following info is printed to the console:

"Anna is a Programmer who earns $100000 in salary"

(See solution in Sakai++, under JavaScript-OOP-Inheritance)

**/\* eslint-disable require-jsdoc \*/**

**"use strict";**

**/\*\***

**\* app.js**

**\*/**

**const person = {**

**name: "",**

**dob: "",**

**age: -Infinity,**

**getName: function() {**

**return this.name;**

**},**

**setName: function(newName) {**

**this.name = newName;**

**}**

**};**

**const john = Object.create(person);**

**john.setName("John");**

**john.dob = new Date(1998,12,10);**

**john.age = 20;**

**console.log(`Name is ${john.getName()}`);**

**console.log(`${john.getName(0)}'s age is ${john.age}`);**

**const employee = Object.create(person);**

**employee.salary = 0.00;**

**employee.hireDate = new Date();**

**employee.doJob = function(jobTitle) {**

**console.log(`${this.name} is a ${jobTitle}**

**who earns $${this.salary}`);**

**};**

**const anna = Object.create(employee);**

**anna.setName("Anna");**

**anna.dob = new Date(2000,11,9);**

**anna.age = 18;**

**anna.salary = 125000.00;**

**anna.hireDate = new Date(2019,3,17);**

**anna.doJob("Programmer");**

1. Write a JavaScript function named, findOdds, that takes an array of numbers, as input, and returns an array containing only the odd numbers that are present in the input array. E.g. findOdds(1,2,3) should return [1,3], findOdds(1,2,3,4,5) should return [1,3,5] and so on.

**function findOdds(...array) {**

**let a = [];**

**for(let i=0; i<array.length;i++) {**

**if(array[i]%2===0) a.push(array[i]);**

**}**

**return a;**

**}**

1. Write code for an object named, Account, as a JavaScript object literal, with the following specifications:

Account properties:

- accountNumber (set the default value to an empty string)

- accountType (set the default value to an empty string)

- balance (set the default value to 0.0)

- status (set the default value to an empty string)

Account methods:

Include the following methods:-

- create (This method takes as input, data for accountNumber, accountType, balance and status, which it then assigns/sets to the respective Account object properties; and then it simply prints the message, "New Account, AC-0001, created successfully", where AC-0001 is the accountNumber passed to the method)

- toString (Returns a string representation of the Account object, in JSON format)

1. By applying the **Revealing Module Pattern**, re-implement code for the Account object described in Q6 above. Implement your code such that the accountNumber, accountType, balance and status properties are encapsulated as **private data fields**. Then add the following public getter/setter methods for some of the properties:

getAccountNumber - Returns the value of the accountNumber property,

getAccountType - Returns the value of the accountType property,

getBalance - Returns the value of the balance property,

getAccountStatus - Returns the value of the accountStatus property,

setAccountStatus - Takes as input parameter, accountStatus, which it assigns to the accountStatus property.

Also make the *create* and *toString* methods, public.

1. An object named, tomsCheckingAccount, IS-An Account. Applying JavaScript inheritance, write code that creates this object (tomsCheckingAccount), using the Object.create(...) method; and assigns to it the following values:

**tomsCheckingAccount property values**:

accountNumber - AC-0099",

accountType - "Checking",

balance - $16,595.99,

status - "Open"

Then, add a new method named, payFees, to this tomsCheckingAccount object, which takes as input, as fee amount and deducts it from the account's balance.

1. Using the **Constructor function** way, write code for an object named, Account, based on the specification given in Q6 above.
2. Using HTML, CSS and JavaScript, implement a mini-BankingApp as a single-page web application, as shown in the UI screenshot below, with the following features and functionalities: