**COMP 3540 Advanced Web Design and Programming**

Final Exam – Fall 2017

**Two hours**

Student Name: Student Number:

Note 1. Do not include any unnecessary explanations or guesses in your answers. They will get **negative** marks.

Note 2. Answer clearly, completely, and correctly. In general partial marks will **not** be given.

1. (9 marks) Fill in the blanks in each of the following.
2. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is defined as the set of local variables in a function that are kept alive after the function has returned.
3. All JavaScript objects inherit the properties and functions (or also called methods) from their \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ that is created using an object constructor function.
4. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator removes a given property from an object. (Note that the operator is a keyword, not a symbol.)
5. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a web app that interacts with the user by dynamically rewriting the current page rather than load entire new pages from a server.
6. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a very common data format used for asynchronous client-server communication, including as a replacement for XML in some AJAX-style systems.
7. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a sequence of characters that define a search pattern.
8. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ function is any function that can be used to map data of arbitrary size to data of fixed size that cannot be easily reversed to the original data.
9. In a browser, the window.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ object is used to store data with no expiration date.
10. The PHP super global array, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, is used for uploading files.
11. (3 marks) Complete the following code that directs HTTP channel to a secure channel.

<?php

?>

1. (3 marks) Write the outputs from the following code.

var person = (function(n, a)

var name = n;

var age = a;

return {

get\_name: function() { return name; },

set\_name: function(new\_name) { name = new\_name; },

get\_age: function() { return age; },

set\_age: function(new\_age) { age = new\_age; }

};

}) ('John', 21);

console.log(person.name);

person.name = 'Tom';

console.log(person.get\_name());

console.log(person.name);

person.set\_name('Bob');

console.log(person.get\_name());

console.log(person.name);

1. (6 marks) Complete the following code that

* Displays the data sent from a server script into an iframe, and
* When the iframe is fully loaded, displays ‘Loaded’ using alert().

<form method='get' action='iframe-exercise.php' >

<input type='text' name='command'>

<input type="submit" value="Submit">

</form>

<iframe width='400px' height='100px' >

</iframe>

<script>

</script>

1. (8 marks) Complete the following simple Memo app code that

* Has three buttons
  + ‘Keep it!’ – Keep a memo with a title
  + ‘Display all’ – Display all memos into a <div> of id='memos'
  + ‘Delete all’ – Delete all memos
* Note. You should use just one local storage item for all the memos.

<label for='memo-title'>Title: </label> <input id='memo-title' placeholder='memo title'>

<br>

<label for='memo-content'>Memo: </label>

<textarea id='memo-content' cols='50' placeholder='memo content'></textarea><br>

<div id='memos'>Memos = </div>

<button id='memo-keepit' type='button'>Keep it!</button>

<button id='memo-display' type='button'>Display all!</button>

<button id='memo-remove' type='button'>Remove all!</button>

<script>

$('#memo-keepit').click(function() {

var db = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_('db-memo'); // retrieve the db

// convert the db value to an object

if (db == null) db = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // empty linear array

else db = JSON.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(db);

// construct an object for a new memo; { title:..., ...:... }

var obj = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // empty object

obj\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ = $('#memo-title').\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(); // the memo title

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // and content

db.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(obj); // push the object into db

// convert db to a string and save it into the local storage

db = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(db);

localStorage.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, db);

});

$('#memo-display').click(function() {

var db = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_('db-memo'); // retrieve the db

$('#memos').\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(db); // display the db

});

$('#memo-remove').click(function() { // remove the db from the local storage

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

});

</script>

1. (6 marks) Complete the following code to move an image between two <div> elements.

<style>

#dd-div1, #dd-div2 {

width: 100px; height: 50px;

margin: 10px; padding: 10px;

border: 1px solid black;

}

</style>

<script>

function dragOver(eObj) {

eObj.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_();

}

function dragStart(eObj) {

eObj.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_("dd-id", eObj.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.id);

}

function drop(eObj) {

eObj.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_();

var id = eObj.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_);

eObj.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(document.getElementById(id));

}

</script>

<div id="dd-div1" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_="dragOver(event)" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_="\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_">

<img src="tru.png" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_="true" ondragstart=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

id="dd-image" width="40" height="40">

</div>

<div id="dd-div2" >

</div>

1. (4 marks) Write a regular expression used for complex passwords. The regular expression should support patterns that

* Have minimum 6 characters.
* Start with an uppercase alphabet
* End with a digit
* Include at least one special character from !, @, #, $, %, ^, &, \*, -

1. (12 marks) Complete the following code that

* Reads a query from the user
* Sends the query to a server script
* Displays the data sent back into an <div> of id=‘search-result’.
* Note 1. Two input values must be read and sent.
* Note 2. You should use jQuery post function.
* Note 3. The data sent back from the server script is a JSON string of linear array of associative arrays.
* Note 4. The output to the user should be a table. You can use the function construct\_table(). The table should include column names.

<div>

<input type='text' placeholder='Enter a category'

>

<br>

<input type='text' placeholder='Enter a search term'

>

<br>

<div id='search-result'><div>

</div>

<script>

function construct\_table(data) {

var data = JSON.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(data); // You can assume there is at least one stored.

var str = '<table>';

str += '</table>';

return str;

}

// \*\*\* continue to the next page \*\*\*

var controller = "//cs.tru.ca/truqa/controller.php"; // the server script that handles

// the user query

var category;

var search\_term;

</script>