## 10% - //HTML: code used inside the <body>

#### Choose any TEN options

\_\_\_\_ Contains one or more linked images: <img src='http://...'>
\_\_\_\_ Contains a button: <button>
\_\_\_\_ Contains a text input line: <input>
\_\_\_\_ Contains a text input area: <textarea>
\_\_\_\_ Contains a link to another document: <a href=''>
\_\_\_\_ Contains a header: <h1> | <h2> | <h3> | <h4> | <h5> | <h6>
\_\_\_\_ Contains a paragraph: <p>
\_\_\_\_ Contains a line break: <br>
\_\_\_\_ Contains strong or emphasized text: <strong> | <em>
\_\_\_\_ Contains a span or div: <span> | <div>
\_\_\_\_ Contains a list of items: <li>
\_\_\_\_ Contains a table: <table>
\_\_\_\_ Contains HTML unicode characters: &#74;
\_\_\_\_ Contains an HTML canvas: <canvas>
\_\_\_\_ Contains Scalable Vector Graphics: <svg>
\_\_\_\_ Contains a playable mp3 file: <audio>

\_\_\_\_ *Contains other HTML as approved by instructor*

## 10% - //CSS: code must be used inside <style>

#### Choose any TEN options

\_\_\_\_ Apply a style to a tag: p { ... }
\_\_\_\_ Apply a style to an ID: #exampleId { ... }
\_\_\_\_ Apply a style to a class: .exampleClass { ... }
\_\_\_\_ Positions something using % instead of px: left: 30%;
\_\_\_\_ Sets the family of text: font-family: ...
\_\_\_\_ Sets the border: border: ...
\_\_\_\_ Sets the color of text: color: ...
\_\_\_\_ Sets the background color of text: background-color: ...
\_\_\_\_ Sets display or visibility: display: ... | visibility: ...
\_\_\_\_ Sets the size of text: font-size: ...
\_\_\_\_ Sets the weight of text: font-weight: ...
\_\_\_\_ Sets the style of text: font-style: ...
\_\_\_\_ Sets the width or height of an element: width: ... | height: ...
\_\_\_\_ Sets the positioning of an element: position: ...
\_\_\_\_ Positions an element using measurements: top: ... | left: ...

\_\_\_\_ Sets the text alignment: text-align: ... | vertical-align: ...
\_\_\_\_ Sets opacity: opacity: ...
\_\_\_\_ Sets the stacking order: z-index: ...
\_\_\_\_ Sets CSS animation: @keyframes ...

\_\_\_\_ Links to a web font: @font-face

\_\_\_\_ *Uses other CSS as approved by instructor*

## 5% - //INIT: code used inside the <script>

#### Choose any FIVE

\_\_\_\_ Links to jQuery
\_\_\_\_ Uses numerical variables: exampleNumber = 42;
\_\_\_\_ Initializes a numerical variable as 0: exampleNumber = 0;
\_\_\_\_ Uses string variables: exampleStr = 'Hello!';
\_\_\_\_ Initializes a string variable as: exampleStr = '';
\_\_\_\_ Uses an array: exampleArr = [ 'Hello', ' world!' ];
\_\_\_\_ Initializes an array as []: exampleArr = [];
\_\_\_\_ Uses an object: exampleObj = { language: 'English', greeting: 'Hi!' };
\_\_\_\_ Initializes an object as {}: exampleObj = {};
\_\_\_\_ Uses a two dimensional array: exampleArr = [ [ 1, 2, 3 ], [ 4, 5, 6 ] ];
\_\_\_\_ Uses a complex data structure (eg: an array of objects):

exampleArr = [

{ language: 'English', greeting: 'Hello!' },

{ language: 'German', greeting: 'Guten Tag!' }

];
\_\_\_\_ *Uses other variable structures as approved by instructor*

## 5% - //INPUT: trigger functions and read information

#### Choose any FIVE options

\_\_\_\_ Trigger things by clicking on an HTML element:

document.getElementById(…).onclick = exampleFunction;

\_\_\_\_ Trigger things (like the whole page) when they finish loading:

document.body.onload = exampleFunction()

\_\_\_\_ Trigger things by changing the contents of an input line or area:

document.getElementById(…).onchange = exampleFunction;

\_\_\_\_ Read information from an input line:

document.getElementById( 'inputId' ).value

\_\_\_\_ Read information from the keyboard:

window.onkeydown = function ( keyEvent ) { ... }

\_\_\_\_ Trigger a function by mousing over or mousing out:

document.getElementById(…).onmouseover =exampleFunction;

\_\_\_\_ Trigger a function by moving the mouse:

document.onmousemove = function ( mouseEvent ) { ... }

\_\_\_\_ Trigger a function by pressing the mouse button:

document.onmousedown = function () { ... }

\_\_\_\_ Read the height or width of the window:

window.innerWidth | window.innerHeight

\_\_\_\_ Read the current position of an element:

document.getElementById('inputId').offsetTop | .offsetLeft

\_\_\_\_ Read the current position of the mouse pointer:

window.clientX | window.clientY

\_\_\_\_ Turning OFF the click or mouse functions of an element:

document.getElementById('exampleId').onmouseover = ''

\_\_\_\_ The program reads the time: new Date()
\_\_\_\_ The user uploads a file: var reader = new FileReader();
\_\_\_\_ Gets data from other websites: $.get( 'https://drapak.ca' );

\_\_\_\_ *Reads other input as approved by instructor*

## 15% - //PROCESS: code inside the <script>

#### Choose any FIFTEEN options (some can be used more than once)

\_\_\_\_ \_\_\_\_ Uses a singular if statement: if ( exampleVar == 1 ) { ... }
\_\_\_\_ Uses an if...else if...else chain:

if ( exampleVar == 1 ) { ... }

else if ( exampleVar == 2 ) { ... }

else { ... }

\_\_\_\_ \_\_\_\_ Uses a for loop: for ( i in exampleArr ) { ... }
\_\_\_\_ Uses a nested loop
\_\_\_\_ \_\_\_\_ \_\_\_\_ Uses string addition: 'Hello' + ' world!'
\_\_\_\_ Uses a split function: var newArr = exampleVar.split(' ');
\_\_\_\_ Uses a join function: var newStr = exampleArr.join(', ');
\_\_\_\_ \_\_\_\_ \_\_\_\_ Uses arithmetic functions: 5 + 6 - 7 \* 3 / 4
\_\_\_\_ Uses rounding functions: Math.round() | .floor() | .ceil()
\_\_\_\_ Generates random numbers: Math.random()
\_\_\_\_ \_\_\_\_ \_\_\_\_ Uses .Math (powers, trig, etc.): Math.pow() | Math.PI

\_\_\_\_ Extracts numbers from a string: parseInt() | parseFloat()
\_\_\_\_ Uses other array functions: exampleArr.pop() | .shift() | .unshift()
\_\_\_\_ Uses animation callbacks:

$('#exampleId').animate(

{ top: '100px' },

{

duration: 1000,

 easing: 'linear',

 complete: function () { ... }

}

);

\_\_\_\_ \_\_\_\_ Uses setInterval() or setTimeout()

\_\_\_\_ creates a new element: document.createElement( 'img' )

\_\_\_\_ \_\_\_\_ Uses JSON to exchange data: JSON.parse() | .stringify()

\_\_\_\_ \_\_\_\_ Uses window.requestAnimationFrame()

\_\_\_\_ *Uses other process functions as approved by instructor*

## 5% - //OUTPUT: code inside the <script>

#### Choose any FIVE options

\_\_\_\_ Outputs inside an html tag:

document.getElementById( 'exampleId' ).innerHTML

\_\_\_\_ Adds/removes/changes an image:

document.getElementById( 'exampleId' ).src

\_\_\_\_ Changes the position of an element:

document.getElementById( 'exampleId' ).style.top | .left

\_\_\_\_ Changes the style of an element:

document.getElementById( 'exampleId' ).style.backgroundColor

\_\_\_\_ Uses animation to change position:

$( '#exampleId' ).animate(

{

top: '+=100px',

left: '-=100px',

 },

{

duration: 1000,

easing: 'linear'

}

);

\_\_\_\_ Uses animation to change other style properties:

$( '#exampleId' ).animate(

{ width: '100px' },

{ duration: 1000,

easing: 'linear'

}

);

\_\_\_\_ Creates a pop up window: alert( 'Hi there!' );

\_\_\_\_ Sends data to the developer's console:

console.log( 'in mainProcedure...' );

\_\_\_\_ Plays an audio clip:

document.getElementById( 'audioId' ).play();

\_\_\_\_ appends an element to an id:

document.getElementById( 'outputId' ).appendChild(

 newElement

);

\_\_\_\_ *Uses other output as approved by instructor*

## 20% - //Programming structure

#### Use ALL TEN skills (each skill has double weight)

☐ Uses a function:

var exampleFunction = function () { ... }

☐ Use a function that accepts parameters/arguments as input:

var exampleFunction = function ( exampleVariable ) { ... }

☐ Uses a function that returns information:

return exampleVariable

☐ Uses an object-oriented structure with a method:

var exampleObject = { exampleMethod: function () {...} }

☐ Uses .this inside an object-oriented structure:

this.row = this.row + 1;

☐ Uses an object constructor:

var ExampleConstructor = function ( exampleVar ) {

this.exampleProperty = exampleVar;

};

☐ Uses a separate function/method for initializing data

☐ Uses a separate function/method for reading input

☐ Uses a separate function/method for processing information

☐ Uses a separate function/method for outputting information

☐ ***Optional:*** *Uses other program structure as approved by instructor*

## 20% - //STYLE: Professional programming habits and communication

#### Use ALL TWENTY skills

☐ Correct filename: exam-LastName.html

☐ The author of the program is indicated in the <head>
☐ Comment for the date started and dates modified,

 including what was modified

☐ All CSS code is in the <style> block of the file

☐ The <style> block is in the <head> of the file

☐ The <script> block is at the end of the <body>

☐ Avoids lines longer than 96 characters
☐ Space out mathematical operators: var newScore = oldScore + 30;
☐ Accurately indents code

☐ Comment for describing each //INIT: block, function or method
☐ Comment for describing each //INPUT: function or method
☐ Comment for describing each //PROCESS: function or method
☐ Comment for describing each //OUTPUT: function or method

☐ Descriptive comments for each loop
☐ Descriptive comments for each if statement

☐ Uses lower-case for HTML tags

☐ Uses camelCase for #id and .class names
☐ Uses camelCase for variable and function names
☐ Uses descriptive, full language variable names
☐ Uses descriptive, full language function and method names

## 10% - //CHECKLIST

#### The last day of class will be devoted to completing this checklist

☐ x 5 Completes and hands in an exam skills checklist

☐ x 5 Uses accurate line numbers to where a skill is used

## Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Advice

* Keep your classmates informed about what you are doing.
* Get things working in bite-sized chunks.
* Use console logs to tell you where you are in your code.
* Use console logs to tell you what values your variables contain.
* Do CSS work when you are stuck or need a break.
* Plan your data structures carefully before you start.
* Have a clear definition of what you want input and what you want to output. This makes the process much easier.
* Save your work frequently. Submit your in-progress versions.
* When looking online for advice, keep in mind that the quality of publicly available code is iffy. Get me to take a quick look at it...