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: **
**
___ Contains strong or emphasized text: ** | **
___ Contains a span or div: ** | **
___ Contains a list of items: **<li>**
___ Contains a table: **<table>**
___ Contains HTML unicode characters: **J**
___ 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...