

---

## CSCI 3300 Assignment 5

**Total estimated time** for this assignment: **10 hours**

When you see “Richard Ricardo” in the example screen captures, change it to **<your name>**.

When you see “Richard” in the example screen captures, change it to **<your first name>**.

If you do not put **<your name>** / **<your first name>** in the above mentioned fields, you will get **0 points** for the question(s).

**No two students** should submit webpages with exactly the same code, or same content, or same layout, or same color combination. If found, **both** students will get **0 points**.

Create a folder on your hard disk, name the folder **lastname\_firstname\_assignment5**. Save all the files from this assignment in this folder.

Create the following subfolders (in the folder lastname\_firstname\_assignment5): **q1, q2, q3, q4**.

As a result, you should have the following folder (directory) structure for this assignment: (-2 points if wrong)

- lastname\_firstname\_assignment5\q1\
- lastname\_firstname\_assignment5\q2\
- lastname\_firstname\_assignment5\q3\
- lastname\_firstname\_assignment5\q4\

Use Firefox **Web Console** (Tools > Web Developer > Web Console) to help debugging JavaScript. All html (JavaScript) files must pass JavaScript validation at Firefox Web Console without any error, without any warning (**-2 points for each error, each warning**).

- Turn on your Firefox **Menu Bar** (Hint: right-click).
- Make sure your Firefox is up to date (Help > About Firefox).
- Turn on Web Console when you code (Tools > Web Developer > Web Console).
- Fix any error/warning immediately when you see them. Do NOT accumulate errors.

When you view page source in a web browser, **<!DOCTYPE html>** must be at the top of every page. In other words, all pages must be written in HTML5. (**-20 points** if not)

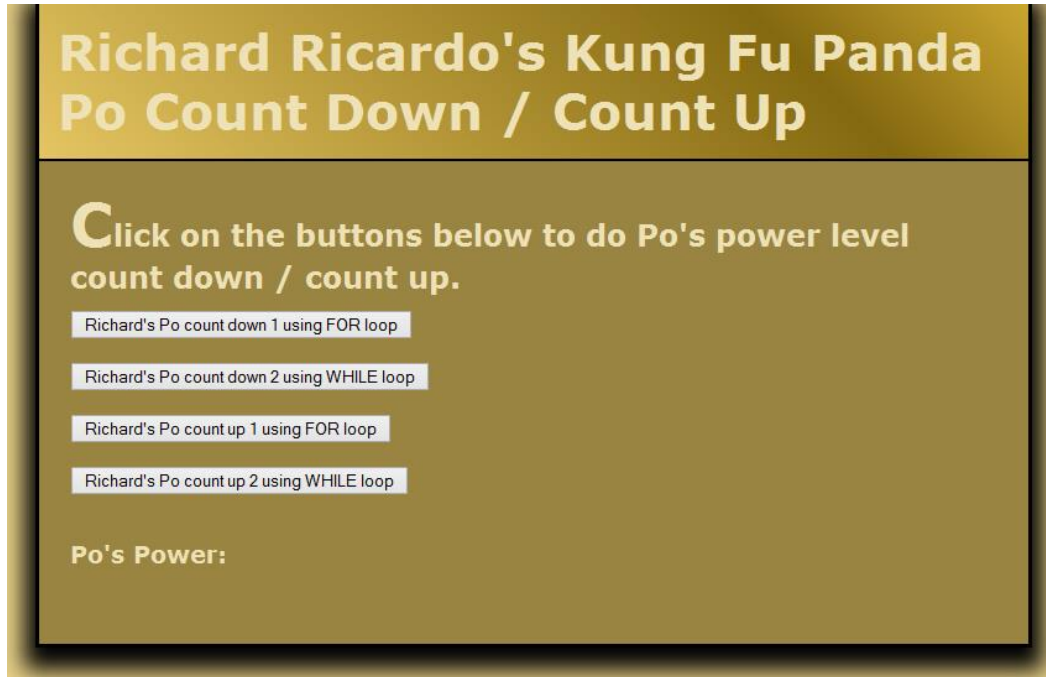
**Question 1** – JavaScript Chapter 4 (25 points)

**Estimated time: 2 hours**

- Save question 1 files in subfolder “**lastname\_firstname\_assignment5q1\**”. (-20 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 5 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using “**<your name>**’s Kung Fu Panda Po Count Down / Count Up” as the page title. Save the page as **index.htm**. Document the html file with html and JavaScript comments. (1 point each, total 3 points)
- Allow the user to click on one of the four buttons.
  - Use **for** loop to achieve button 1’s effects (4 points)
  - Use **while** loop to achieve button 2’s effects (4 points)
  - Use **for** loop to achieve button 3’s effects (4 points)

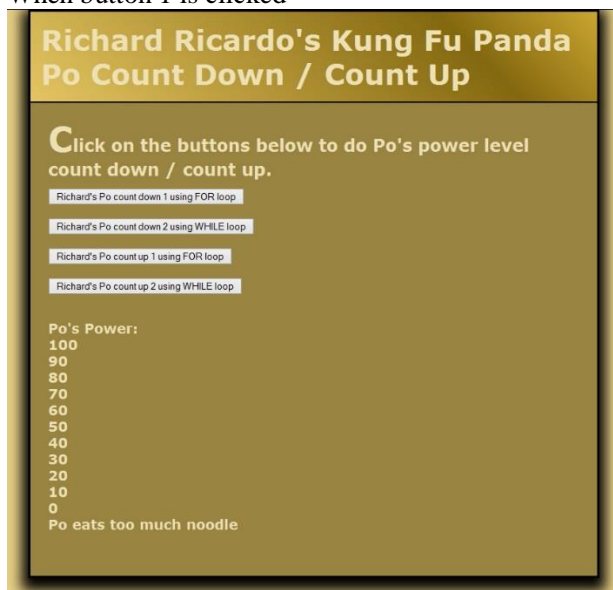
- Use **while** loop to achieve button 4's effects (4 points)
- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. You can use **the same (or similar) css file(s)** to format all questions. (1 point each, total 2 points)

Example: Initial Page

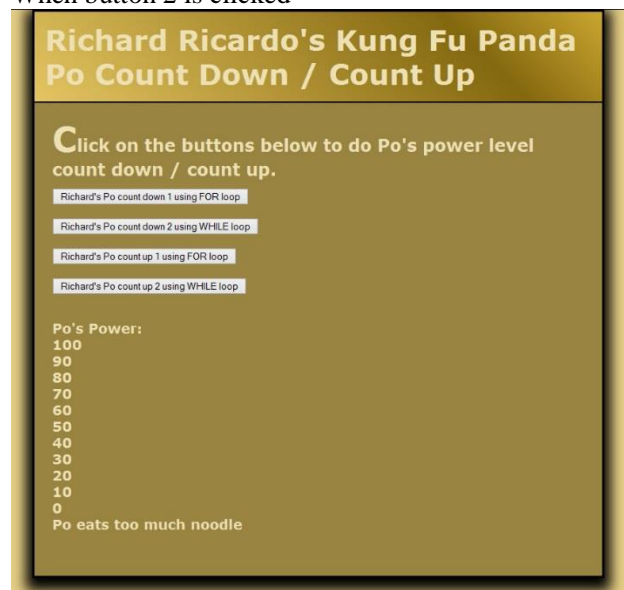


Example: Output

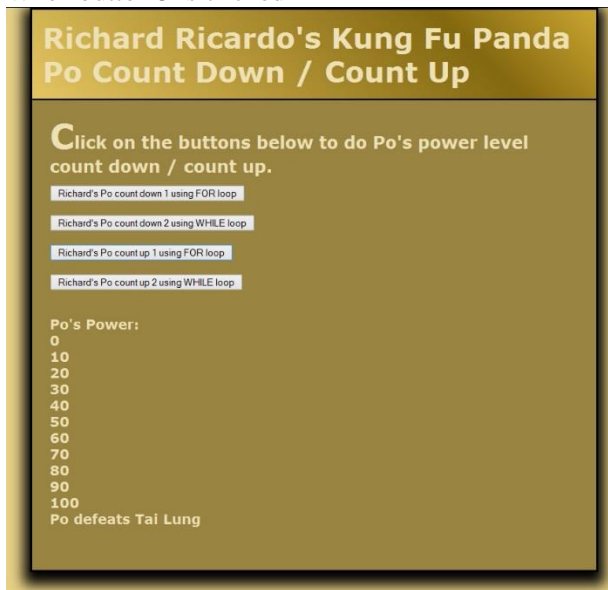
When button 1 is clicked



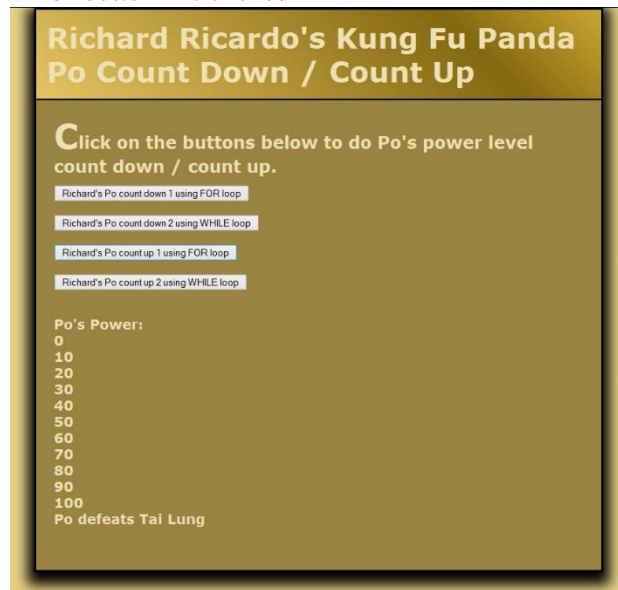
When button 2 is clicked



When button 3 is clicked



When button 4 is clicked



**Question 2** – JavaScript Chapter 4 (25 points)

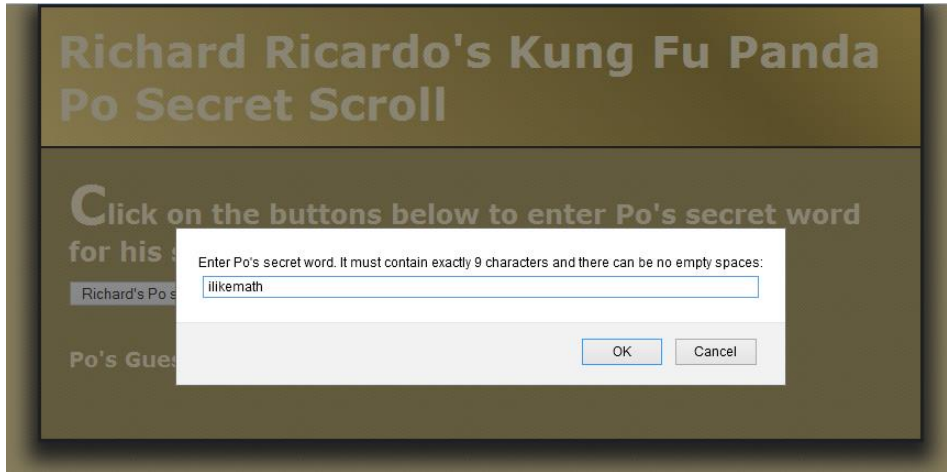
**Estimated time: 3 hours**

- Save question 2 files in subfolder “**lastname\_firstname\_assignment5\q2\**”. (-20 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 4 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using “<your name>’s Kung Fu Panda Po Secret Scroll” as the page title. Save the page as **index.htm**. Document the html file with html and JavaScript comments. (1 point each, total 3 points)
- Write a program that asks the user for Po’s secret word.
  - The secret word must have exactly **nine characters** and **may not include spaces**. (8 points)
  - All other keyboard characters are allowed.
  - A **loop (while or do...while)** should prompt the user to re-enter another secret word until **both** these conditions are met. (10 points for loop logic, 2 points for output format)
- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. (1 point each, total 2 points)

Example: Initial Page



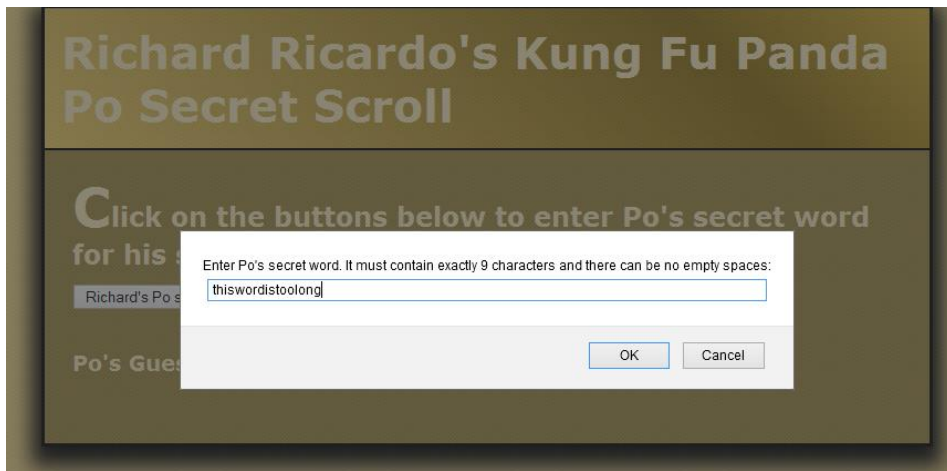
Example: Input q2a



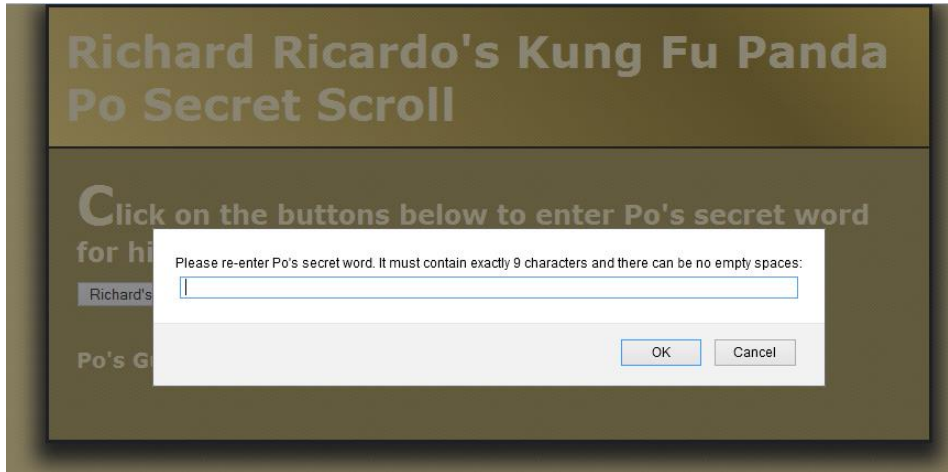
Example: Output q2a



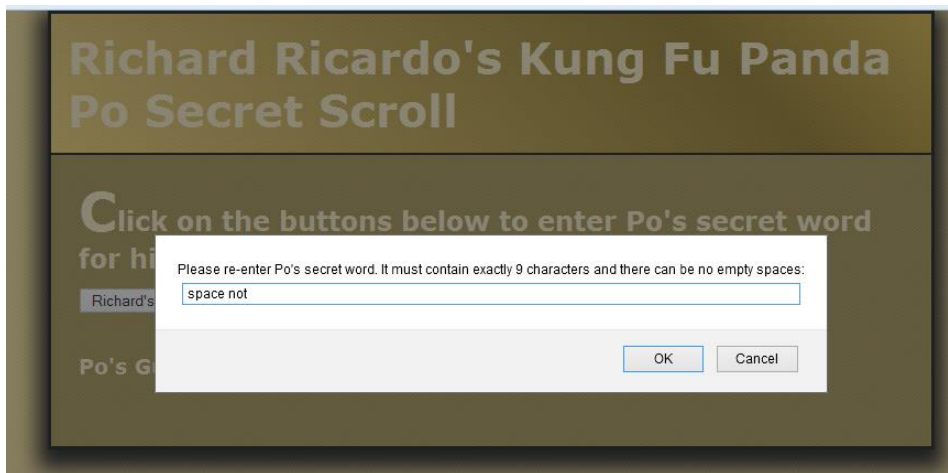
Example: Input q2b



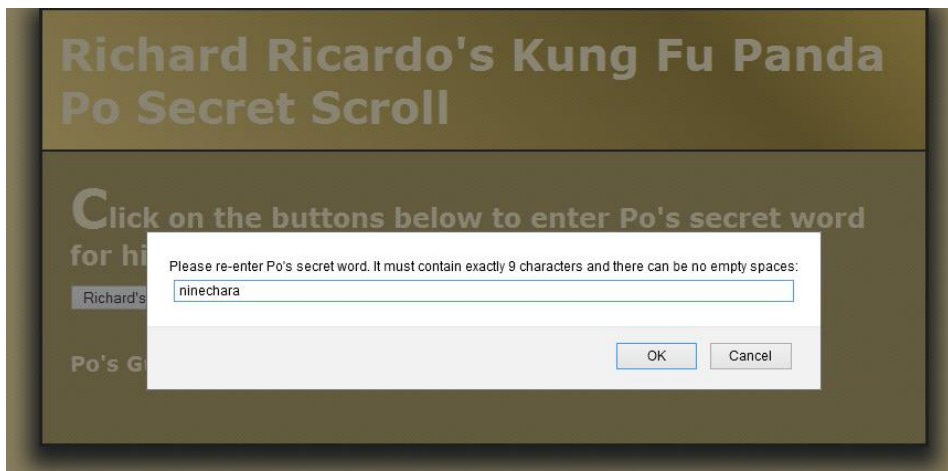
Example: Output q2b



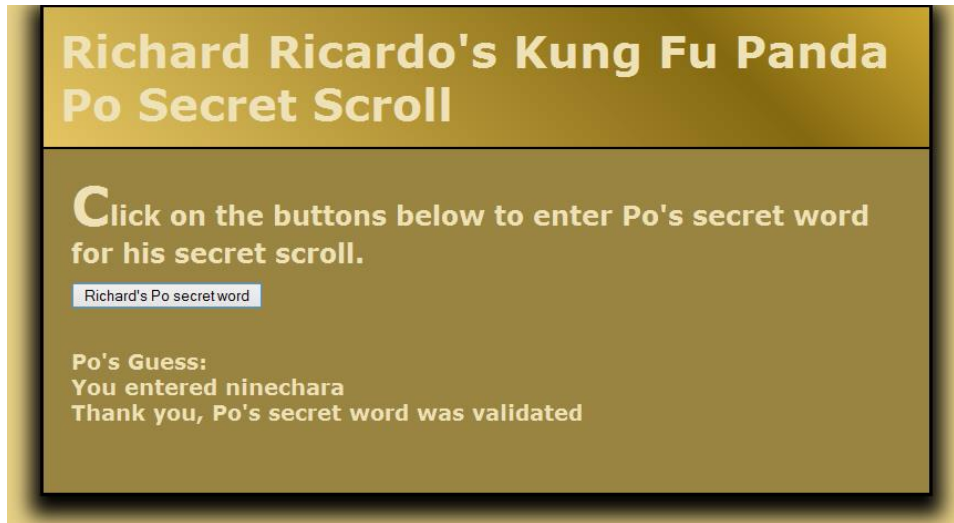
Example: Input q2c



Example: Output q2c and Input q2d



Example: Output q2d



**Question 3** – JavaScript Chapter 4 & 5 (25 points)

**Estimated time: 2 hours**

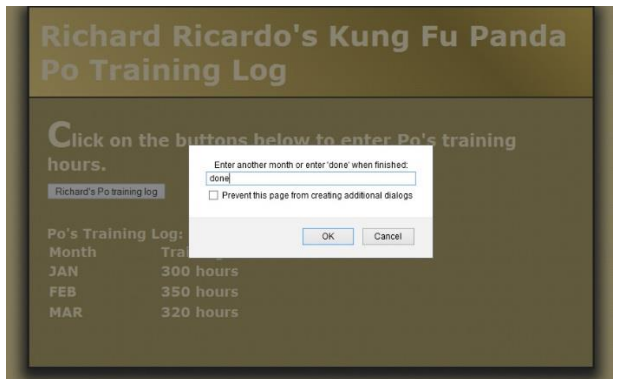
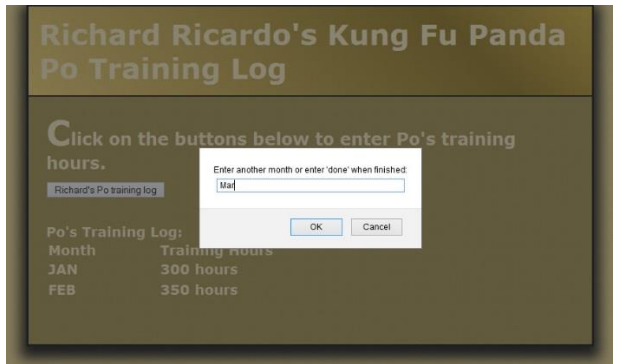
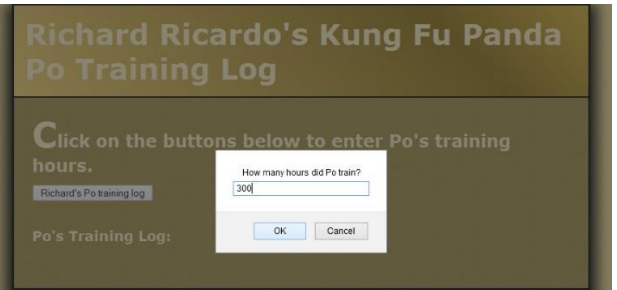
- Save question 3 files in subfolder “**lastname\_firstname\_assignment5\q3**”. (-20 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 4 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using “<your name>’s Kung Fu Panda Po Training Log” as the page title. Save the page as **index.htm**. Document the html file with html and JavaScript comments. (1 point each, total 3 points)
- Write a program that allows the user to enter Po’s training hours for each month.
  - Allow user to enter as many data set as desired (You must use **do...while** loop, 10 points).
  - The output should look like the example output shown. (2 points for output format)
  - The program should also calculate Po’s **total** training hours. (5 points)
- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. (1 point each, total 2 points)

Example: Initial Page





Example: Input q3



Example: Output q3



**Richard Ricardo's Kung Fu Panda Po Training Log**

**C**lick on the buttons below to enter Po's training hours.

Richard's Po training log

**Po's Training Log:**

Month	Training Hours
JAN	300 hours
FEB	350 hours
MAR	320 hours

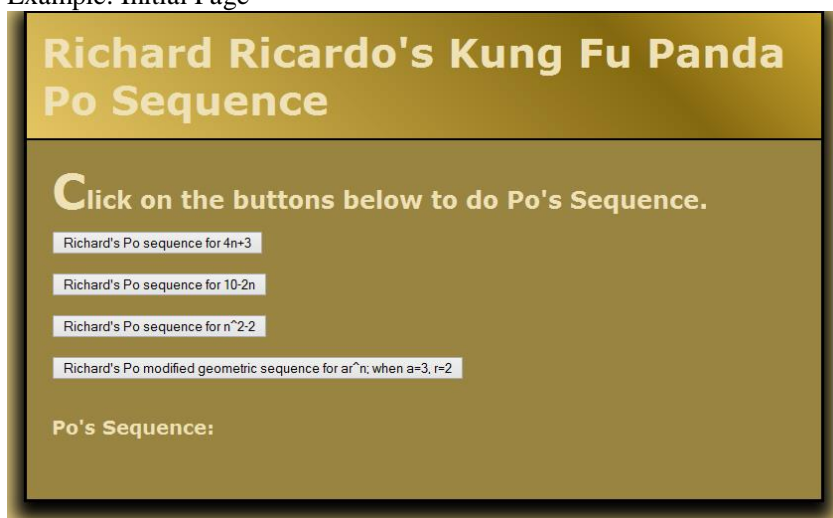
Po trained a total of 970 hours.

**Question 4** – JavaScript Chapter 4 & 5 (25 points)

**Estimated time: 3 hours**

- Save question 4 files in subfolder “**lastname\_firstname\_assignment5\q4**”. (-20 points if no JavaScript)
- Create a web page that displays the default information (example shown). (1 point each, total 4 points)
- The initial page and related outputs should look like the examples shown below.
- Create your page using “<your name>’s Kung Fu Panda Po Sequence” as the page title. Save the page as **index.htm**. Document the html file with html and JavaScript comments. (1 point each, total 3 points)
- Allow the user to click on one of the four buttons. (4 points each for logic, total 16 points)
  - You can (must) choose to use **for** loop, **while** loop or **do..while** loop.
  - The sequences generated and the output should look like the example output shown. (1 point each for output, total 4 points)
- Create a css file named **style.css** to format index.htm by creating your own layout (no two students should have the same layout). Use css comments to document the css program. You can use **the same (or similar) css file(s)** to format all questions. (1 point each, total 2 points)

Example: Initial Page



**Richard Ricardo's Kung Fu Panda Po Sequence**

**C**lick on the buttons below to do Po's Sequence.

Richard's Po sequence for  $4n-3$

Richard's Po sequence for  $10-2n$

Richard's Po sequence for  $n^2-2$

Richard's Po modified geometric sequence for  $ar^n$ , when  $a=3, r=2$

**Po's Sequence:**



## Example: Output

When button 1 is clicked

**Richard Ricardo's Kung Fu Panda Po Sequence**

Click on the buttons below to do Po's Sequence.

Richard's Po sequence for  $4n-3$

Richard's Po sequence for  $10-2n$

Richard's Po sequence for  $n^2-2$

Richard's Po modified geometric sequence for  $a^n$ , when  $a=3, r=2$

Po's Sequence:

When n=1;	$4n+3 = 4 \times 1 + 3 = 7$
When n=2;	$4n+3 = 4 \times 2 + 3 = 11$
When n=3;	$4n+3 = 4 \times 3 + 3 = 15$
When n=4;	$4n+3 = 4 \times 4 + 3 = 19$
When n=5;	$4n+3 = 4 \times 5 + 3 = 23$
When n=6;	$4n+3 = 4 \times 6 + 3 = 27$
When n=7;	$4n+3 = 4 \times 7 + 3 = 31$
When n=8;	$4n+3 = 4 \times 8 + 3 = 35$
When n=9;	$4n+3 = 4 \times 9 + 3 = 39$

When button 2 is clicked

**Richard Ricardo's Kung Fu Panda Po Sequence**

Click on the buttons below to do Po's Sequence.

Richard's Po sequence for  $4n-3$

Richard's Po sequence for  $10-2n$

Richard's Po sequence for  $n^2-2$

Richard's Po modified geometric sequence for  $a^n$ , when  $a=3, r=2$

Po's Sequence:

When n=1;	$10-2n = 10-2 \times 1 = 8$
When n=2;	$10-2n = 10-2 \times 2 = 6$
When n=3;	$10-2n = 10-2 \times 3 = 4$
When n=4;	$10-2n = 10-2 \times 4 = 2$
When n=5;	$10-2n = 10-2 \times 5 = 0$
When n=6;	$10-2n = 10-2 \times 6 = -2$
When n=7;	$10-2n = 10-2 \times 7 = -4$
When n=8;	$10-2n = 10-2 \times 8 = -6$
When n=9;	$10-2n = 10-2 \times 9 = -8$

When button 3 is clicked

**Richard Ricardo's Kung Fu Panda Po Sequence**

Click on the buttons below to do Po's Sequence.

Richard's Po sequence for  $4n-3$

Richard's Po sequence for  $10-2n$

Richard's Po sequence for  $n^2-2$

Richard's Po modified geometric sequence for  $a^n$ , when  $a=3, r=2$

Po's Sequence:

When n=1;	$n^*n-2 = 1 \times 1 - 2 = -1$
When n=2;	$n^*n-2 = 2 \times 2 - 2 = 2$
When n=3;	$n^*n-2 = 3 \times 3 - 2 = 7$
When n=4;	$n^*n-2 = 4 \times 4 - 2 = 14$
When n=5;	$n^*n-2 = 5 \times 5 - 2 = 23$
When n=6;	$n^*n-2 = 6 \times 6 - 2 = 34$
When n=7;	$n^*n-2 = 7 \times 7 - 2 = 47$
When n=8;	$n^*n-2 = 8 \times 8 - 2 = 62$
When n=9;	$n^*n-2 = 9 \times 9 - 2 = 79$

When button 4 is clicked

**Richard Ricardo's Kung Fu Panda Po Sequence**

Click on the buttons below to do Po's Sequence.

Richard's Po sequence for  $4n-3$

Richard's Po sequence for  $10-2n$

Richard's Po sequence for  $n^2-2$

Richard's Po modified geometric sequence for  $a^n$ , when  $a=3, r=2$

Po's Sequence:

When n=1;	$3 \times 2 = 6$
When n=2;	$6 \times 2 = 12$
When n=3;	$12 \times 2 = 24$
When n=4;	$24 \times 2 = 48$
When n=5;	$48 \times 2 = 96$
When n=6;	$96 \times 2 = 192$
When n=7;	$192 \times 2 = 384$
When n=8;	$384 \times 2 = 768$
When n=9;	$768 \times 2 = 1536$

**Important:**

1. If you do not put **<your name>** / **<your first name>** in the above mentioned fields (as shown in the examples), you will get **0 points** for the question(s).
2. **No two students** should submit webpages with exactly the same cod, or the content, or same layout, or same color combination. If found, both students will get **0 points**.
3. When you view page source in a web browser, **<!DOCTYPE html>** must be at the top of every page. In other words, all pages must be written in HTML5. (**-20 points** if not)
4. All html files must pass html validation at <http://validator.w3.org/> without any **error/warning** (with only 2 warnings). Use the validator's "File Upload" tab to check each file. (**-2 points** for each error/warning, **2 warnings allowed**)
5. All css files must pass css validation at <http://jigsaw.w3.org/css-validator/> without any **error/warning**. (**-2 points** for each error/warning)

6. If your html file contains any css component, your html file must pass both html validation (3 above), and css validation (4 above) without any error.
7. If your files do not pass the validations, **2 points will be deducted** for **each error** (and each JavaScript warning) found.
8. Document (comment) your html files (`<!-- -->`), css files (`/* */`), and JavaScript files (`/* */`). (**-1 point** for each file with insufficient comments)

#### Submission instructions:

- You need to test the above document(s) in your web browser.
- Do screen capture(s) of the **initial page** and the related **output(s)**. Use any graphic editing software (e.g. Microsoft Paint, Adobe Fireworks, GIMP, or Microsoft Expression Design etc) to cut out the browser output (from the screen capture), paste them into a word document.
- For this assignment, you only need to do screen capture(s) of the **initial page** and the related **output(s)**, you do not need to do screen capture(s) of the input pages. Provide **2 different test cases** for each question. In other words, for **each question**, you need to have **1 initial page** screen capture and **2 related output(s)** screen captures. (-50 points for no test cases, -5 points for only 1 test case)
- Do screen capture(s) of html validation results and css validation results, cut and paste them into the word document. You do not need to do screen capture(s) of JavaScript Lint validation results.
- Save the word document as a pdf file.

You need to submit the following:

1. A pdf file containing the screen capture(s) of the web browser output (all html pages) and the screen capture(s) of all html validation results (from <http://validator.w3.org/>), and css validation results (from <http://jigsaw.w3.org/css-validator/>), name the file **lastname\_firstname\_assignment05.pdf**.
2. All html file(s), css file(s), and other related files (e.g. image files). Zip your file folder (lastname\_firstname\_assignment5) into a single zip file (or rar file) **lastname\_firstname\_assignment05.zip**. In the above example, the zip file should contain the following files and subfolders. If there is any image, there should be a \images\ subfolder.

- lastname\_firstname\_assignment5\q1\index.htm
- lastname\_firstname\_assignment5\q1\style.css
- lastname\_firstname\_assignment5\q2\index.htm
- lastname\_firstname\_assignment5\q2\style.css // you may put style.css in a subfolder
- lastname\_firstname\_assignment5\q3\index.htm
- lastname\_firstname\_assignment5\q3\style.css
- lastname\_firstname\_assignment5\q4\index.htm
- lastname\_firstname\_assignment5\q4\style.css

Please submit an electronic copy (the above mentioned **two files**: .pdf and .zip) to D2L digital dropbox.

#### Grading guidelines (programming questions):

Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile (run) correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down in a reasonable way (e.g. classes and methods)?
- Style (10%): Is the program **indented** properly? Do variables have **meaningful names**?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a **comment** that identifies the author, the course code, and the program date? Are all the classes, methods and data fields clearly **documented (commented)**? Are unclear parts of code **documented (commented)**? (Some items mentioned may not apply to some languages)

A program that does not compile (run) will get at most **50% of the possible points**.

