

**CSI3140 (Spring-Summer 2018)**  
**ASSIGNMENT 4**

**Due Date: Email your completed assignment directly to TA1 by 14:00 on Tuesday, July 10, 2018.**

---

Instructions:

- This assignment has 2 pages and **must be done individually**. It consists of 7 questions with the mark for each question indicated below, resulting in 75 marks in total.
  - **Late assignments will NOT be accepted: They will receive a grade of zero.**
  - Save your answer to each question in a separate file (or a set of files) and name it (or them) using the following format: **AssignmentNo\_QuestionNo\_YourLastName\_YourStudentID**  
For example, John Smith, whose student ID is 1234567, should save his solution to Question 1 (consisting of 3 files) under the file names: **A4\_Q1\_Smith\_1234567.html**, **A4\_Q1\_Smith\_1234567.js**, and **A4\_Q1\_Smith\_1234567.css**
  - Create a cover page in Word format containing your full name, student ID, course number, and assignment number. Name your cover page as **A4\_CoverPage\_YourLastName\_YourStudentID.docx**. The TA will provide your assignment mark on this cover page and send it back to you (with your answer files if he/she has comments/feedback on them).
  - Zip all your answer files together with your cover page into a single .zip file and name it **A4\_YourLastName\_YourStudentID.zip**, and email that .zip file to the TA by the due time and date above.
  - The subject line of your email should follow the format: **CSI3140\_A4\_YourLastName\_YourStudentID**
  - **Important Note:** Google has recently blocked JavaScript files (files with .js extension) from being attached to an email. They also prevent users from uploading a zip file containing .js files. Thus, you will have to change the .js extension into .txt, in order to email your files to the TA, who then will change the .txt extension back to .js before testing them.
- 

Most of the following questions require JavaScript programming. Your code should be well written (e.g., variables declared, meaningful variable names, easy to understand and no confusing shortcuts, meaningful functions and useful comments, etc.).

1. Create a webpage that enables the user to play the game of 15-Puzzle. There's a 4-by-4 board (implemented as an HTML5 table) for a total of 16 slots. One of the slots is empty. The other slots are occupied by 15 tiles, randomly numbered from 1 through 15. Any tile next to the currently empty slot can be moved into the currently empty slot by clicking on the tile. Your program should create the board with the tiles out of order. The user's goal is to arrange the tiles in sequential order row by row. Using the DOM and the *click* event, write a script that allows the user to swap the positions of the open position and an adjacent tile. Alert the user of illegal moves (i.e., when the user clicks a tile that is not adjacent to the empty slot). Separate your script and CSS

rules (if any) from the HTML5 file. Hint: the *click* event should be specified for each table cell. [15 marks]

2. Modify your solution to Question 1 above to determine when the game is over (i.e., when the user has arranged all the tiles in sequential order row by row), then prompt the user to determine whether to play again. If so, scramble the numbers using the *Math.random* method. [5 marks]
3. Write a script that responds to a click anywhere on an HTML5 page by displaying an *alert* dialog as follows: Display the event name if the user held *Shift* during the mouse click. Display the element name that triggered the event if the user held *Ctrl* during the mouse click. Your HTML5 document should contain a number of different elements (e.g., *p*, *anchor*, *list*, etc.) to demonstrate the working of your script. Separate your script from the HTML5 document. [10 marks]
4. Use CSS absolute positioning, *mousedown*, *mousemove*, *mouseup* and the *clientX/clientY* properties of the *event* object to create a program that allows a user to drag and drop an image. When the user clicks the image, it should follow the cursor until the mouse button is released. Provide an image (e.g., a .jpg file) for this purpose. Separate your script and CSS rules from the HTML5 document. [10 marks]
5. Generate a webpage containing the phrase “HTML5 Canvas”. Create a shadow on that phrase with an *offset-x* of 2px, an *offset-y* of 5px, a *blur* of 6px, and a *text-shadow* color gray. Make your own decision on any other necessary parameters. Use HTML5 canvas element. [10 marks]
6. Generate a webpage on which you will use HTML5 canvas to draw a non-rectangular shape using lines. Then, add a vertical linear gradient to the shape with three color stops. Choose your colors. [10 marks]
7. [15 marks]
  - a) Create an XML document that marks up the nutrition facts for a package of Grandma White’s cookies. A package of cookies has a serving size of 1 package and the following nutritional value per serving: 260 calories, 100 fat calories, 11 grams of fat, 2 grams of saturated fat, 5 milligrams of cholesterol, 210 milligrams of sodium, 36 grams of total carbohydrates, 2 grams of fiber, 15 grams of sugars, and 5 grams of protein. Hint: Your markup should contain elements describing the product name, serving size, calories, sodium, cholesterol, protein, etc. Markup each nutrition fact listed above. [5 marks]
  - b) Write an XSL style sheet for the above XML document that displays the nutritional facts in an HTML5 table. [10 marks]