Austin Peay State University, Tennessee Summer 2012
CSCI 3300: Introduction to Web Development
Dr. Leong Lee

CSCI 3300 Assignment 5

Q1: JS Textbook Page 187, Chapter 3, Project **3-2, CalcHeartRate** [25 points]
Note: For this question, name your file `<my name>CalcHeartRate.htm`. For example, my file would be named: `LeongLeeCalcHeartRate.htm`. Change the page title to `<my name> CalcHeartRate.`
Important: If you do not put `<my name>` to the above mentioned fields (page title and filename), **you will get 0 point for the question.**

Estimated time: 2 hours

Q2: **CalcPopulation** [25 points]
Write a JavaScript web page (program) that will predict the size of a population of organisms. The program should ask for the starting number of organisms, their average daily population increase (as percentage), and the number of days they will multiply. For example, a population might begin with two organisms, have an average daily increase of 50 percent, and will be allowed to multiply for eleven days. The program should use a loop to display the size of the population for each day.
Note: For this question, name your file `<my name>CalcPopulation.htm`. For example, my file would be named: `LeongLeeCalcPopulation.htm`. Change the page title to `<my name> CalcPopulation.`
Important: If you do not put `<my name>` to the above mentioned fields (page title and filename), **you will get 0 point for the question.**

Estimated time: 2 hours

Q3: **SquareJS** [25 points]
Write a JavaScript web page (program) that asks the user for a positive integer no greater than 15. The program should display a square on the screen using the character ‘X’. The number entered by the user will be the length of each side of the square. For example, if the user enters 5, the program should display the following:

```
XXXXX
XXXXX
XXXXX
XXXXX
XXXXX
```

Note: For this question, name your file `<my name>SquareJS.htm`. For example, my file would be named: `LeongLeeSquareJS.htm`. Change the page title to `<my name> SquareJS.`
Important: If you do not put `<my name>` to the above mentioned fields (page title and filename), **you will get 0 point for the question.**

Estimated time: 1 hour

Q4: **Registration** [25 points]
Create a registration form, similar to what you might encounter when registering for an online Web site, including two sections: Personal Information, and Security Information.
In the Personal Information section, add name and e-mail address. Include default text in the name and e-mail text boxes, but write some code that removes the default text from each text box when a user clicks it.
In the Security Information section, add password and password confirmation fields. Write code that ensures that the same value was entered into both fields.
Add Submit and Reset buttons that call submit() and reset() event handler functions when they are clicked. The submit() event handler function should ensure that the user has entered values into each text box, and that the values submitted are not the same as the default text. Submit the form to the `FormProcessor.htm` file (which is an empty html file with no content). Save the document as `<my name>Registration.htm.`
Note: For this question, name your file `<my name>Registration.htm`. For example, my file would be named: `LeongLeeRegistration.htm`. Change the page title to `<my name> Registration.`
Important: If you do not put `<my name>` to the above mentioned fields (page title and filename), **you will get 0 point for the question.**

Estimated time: 3 hours

Submission instructions:
You need to test the above programs (questions) separately, and provide **two test cases** (if applicable) for each program (question). Do a screen capture of the input and related output for each test case. Use any graphic editing software (e.g. Microsoft Paint, Adobe Fireworks, GIMP) to cut out the program input and output (from the screen capture), paste them into a word document under a related question number, save the document as a pdf file.

You need to submit the following:
1. A pdf file containing the screen captures of program input and output of all test cases, name the file **lastname_firstname_assignment05.pdf**.
2. All html (or xht) files containing Javascript. Zip your html (or xht) files into a single zip file (or rar file) **lastname_firstname_assignment05.zip**.

Please submit 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 (comments)?** Are unclear parts of code **documented (comments)?** (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**.