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

CSCI 3300 Assignment 3

Create your own survey (web page) of something you are interested in and would like to have a survey. The survey should contain the following:

1. Text field

2. Text area

3. Radio buttons

4. Check boxes

5. Pull-down menu

6. Scrolling multiple-choice menu

7. A submit button (you do NOT need to process the submitted form).

8. All html files must pass validation at http://validator.w3.org/ without any error (and with only 1 warning). Use the validator's "File Upload" tab to check each file.

Submission instructions:
Use notepad++ to create the document(s). You need to test the above document(s) in your web browser. Do a screen capture of the related browser output. 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 under a related question number, save the document as a pdf file.
Do a screen capture of related W3C validator's **validation results** page (for each html file).

You need to submit the following:
1. A pdf file containing the screen capture(s) of the web browser output and the W3C validator's validation results for each html file, name the file **lastname_firstname_assignment03.pdf**.
2. All html file(s). Use **index.htm** for the home page name. Zip your html file(s) and related multimedia files into a single zip file (or rar file) **lastname_firstname_assignment03.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**.