
APSU Coding Camp #A : Assignment 1

Total estimated time for this assignment: **5 hours**

The web site content must be **related to you**. If the content of the web pages are not related to you, you will get **0 points** for the question(s). Please do **NOT** put any sensitive personal information on the web pages.

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_assignment1**. Save all the files from this assignment in this folder.

Question 1 (40 points)

Create a new file folder **lastname_firstname_assignment1**. Keep created files in this folder. (3 points)

Create your personal homepage using “<**your name**>’s personal homepage” as the page title. (2 points)

Save the page as **index.htm**. (2 points)

Be sure the page contains the following:

1. The web page content must be related to you. (-100 points if not)
2. Has at least 3 paragraphs (p tags). (2 points each, 6 points)
3. Has three headers using (any h1 to h6 tags, examples: hobbies, ambitions, etc.). (2 points each, 6 points)
4. Use em (or i) and strong (or b) tags in a paragraph (p tag) and br tag for a break. (2 points each, 8 points)
5. Use address and blockquote tags. (2 points each, 4 points)
6. Add at least six hyperlinks (four are related to Question 2). (2 points each, 12 points, links must work)
7. Display at least two images. (2 points each, 4 points)
8. Show at least two special characters or symbols like ®. (2 points each, 4 points)
Check http://www.w3schools.com/charsets/ref_html_symbols.asp for the list of symbols.
9. Create a css file named **style.css** to format index.htm and provide a basic layout. Use css comments to document the css program. (3 points)
10. Use HTML comments to identify the author, the course code, and the program date. Use HTML comments to document the program. (3 points each, 6 points)

Question 2 (60 points)

Keep all created files in the same file folder **lastname_firstname_assignment1**. (3 points)

Create your personal web pages. The page titles must include <**your name**>, along with a description relevant to the page. In other words, the page titles must have your first name and your last name. (2 points)

1. The content of the web pages must be related to you. (-100 points if not)
2. You should create at least **4 pages** (example: hobbies, ambitions, etc.). Create hyperlinks from the homepage (Question 1, index.htm) for each of these 4 pages **and a link back** to the homepage. (-4 points for each missing page, -1 point for each missing link, 20 points)
3. Use meaningful file names for these 4 pages. No blanks (white spaces), no special characters in the file names. (2 points for each page name, 8 points)
4. The hyperlink from the homepage to the other pages should be contained in an unordered list. (3 points)
5. One page should contain a nested ordered list. (2 points)
6. Each page should have an image (you may search the web for appropriate images but note the copyright of using them). (2 points each, 8 points)

- Update the css file **style.css** (Question 1), and use it to format the 4 pages. You may also create new css files to format these 4 pages (it is your choice). Use text, font, color, and background properties. Create styles for the nested list as well. Use css comments to document the css program. (2 points each, 12 points; 3 points for css comments)
- Use HTML comments to identify the author, the course code, and the program date. Use HTML comments to document the program. (3 points each, 6 points)

Important:

- 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).
- 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**.
- When you view page source in a web browser, **<!DOCTYPE html>** must be at the top of every html page. In other words, all html pages must be written in HTML5. (**-20 points** if not)
- All html files must pass html validation at <http://validator.w3.org/> without any **error/warning** (with only 2 warnings). (**-2 points** for each error/warning, **2 warnings allowed**)
- All css files must pass css validation at <http://jigsaw.w3.org/css-validator/> without any **error/warning**. (**-2 points** for each error/warning)
- 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.
- Document (comment) your html files (`<!-- -->`), css files (`/* */`), and JavaScript files (`/* */`).

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(s) 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.

You need to submit the following:

- 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_assignment01.pdf**. (missing: -50 points; other file format: -10 points)
- All html file(s), css file(s) and other related file (e.g. image files). Zip your file folder into a single zip file (or rar file) **lastname_firstname_assignment01.zip**. (no zip: -10 points)

Please show the above mentioned **two files**: .pdf and .zip to your camp instructor.

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**.