# CSCI 3300 Assignment 1

Create your personal Homepage using your name as the page title. Be sure it contains the following:

0.  The web page content must be related to you.

1.  Have at least 3 paragraphs (p tags).

2.  Have 2 or 3 header (examples: Hobbies, Ambitions, etc.).

3.  Use em (or i) and strong (or b) tags in a paragraph and br tag for a break

4.  Use address and blockquote tags.

5.  Add at least two hyperlinks.

6.  Display at least two images.

7.  Add audio.

8.  Show at least two special characters or symbols like ®.
    Check http://www.w3schools.com/tags/ref_entities.asp for the list of symbols.

9.  Use HTML comment to identify the author, the course code, and the program date. Use HTML comment to document the program.

**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.

You need to submit the following:
1. A pdf file containing the screen capture(s) of the web browser output, name the file **lastname_firstname_assignment01.pdf**.
2. All html file(s). Use **lastname_firstname_assignment01.htm** for the document name. Zip your html file(s) and related multimedia files into a single zip file (or rar file) **lastname_firstname_assignment01.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**.