

CSCI 4000 Assignment 2

Q1: **Registration** [30 points] - PHP

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.
- In the Security Information section, add password and password confirmation fields.
- Use **html table** (or css) to format the input fields (**into rows and columns**).
- Add Submit button. Save the document as `<my name>Registration.php`.
- Submit the form to the `<my name>RegnDisplay.php` file. `<my name>RegnDisplay.php` should use the `$_POST` array to get all the form fields/inputs (name, e-mail, password, confirmed password etc), and display them.
- If any form input is missing, `<my name>Registration.php` should be shown again.
- If any form input is missing, an error message should also be displayed, informing user the first missing form input.
- Use an external *.css file to format both .php files. Create your own *.css file. Do NOT use the provided *.css file in D2L.

Note: For this question, name your files `<my name>Registration.php` and `<my name>RegnDisplay.php`. For example, my files would be named: `LeongLeeRegistration.php` and `LeongLeeRegnDisplay.php`. Change the page titles to `<my name> Registration` and `<my name> Registration Display`.

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 to 3 hours

Q2: **Average** [30 points] - PHP

Write a PHP web page (program) that asks user to enter five numbers (five text boxes). Include default values in these five textboxes. Use html table (or css) to format the input fields (into rows and columns).

- Add statements to display the current year/month/day/time.
- Add Submit and Reset buttons. Save the document as `<my name>Average.php`.
- Submit the form to the `<my name>ProcessAverage.php` file. `<my name>ProcessAverage.php` should use the `$_GET` array to get all the form fields/inputs (five numbers), and display them.
- Add statements to calculate the average of these five numbers, and display the result (with 2 decimal places).
- If any form input is missing, `<my name>Average.php` should be shown again.
- If any form input is missing, an error message should also be displayed, informing user the first missing form input.
- Use an external *.css file to format both .php files. Create your own *.css file. Do NOT use the provided *.css file in D2L.

Note: For this question, name your files `<my name>Average.php` and `<my name>ProcessAverage.php`. For example, my files would be named: `LeongLeeAverage.php` and `LeongLeeProcessAverage.php`. Change the page titles to `<my name> Average` and `<my name> Process Average`.

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 to 3 hours

Q3: Space Cargo Calculation [40 points] – PHP

<Your name>'s Space Travel Company's main business is to equip spaceships with the following standard cargos. The captains of different spaceships would purchase different amounts of cargo, and pay **<Your name>**'s Space Travel Company according to the price listed below.

Cargo Description	Cargo Unit for Sale	Price per Unit
Fuel Pod	1 Pod	\$125.50
Proton Cannon Ammunition	100 Rounds	\$17.2 (per 100 Rounds)
Jetpack	1 jetpack	\$99.00
Oxygen	1 tank	\$50.40

Write a PHP web page (program) `<my name>SpaceCargo.php` that

- Display your company name “**<Your full name>**'s Space Travel Company”
- Read a three-digit spaceship identification number, captain's name, and cargo amount for each of the four cargo categories. Design, format the mentioned using an **html table** (or css) into rows and columns.
- Submit the form to a second file `<my name>ProcessCargo.php`.
- Compute the cost for each of the four cargo categories.
 - The cost for each cargo category **MUST** be computed in a function. There will be only **one function** to calculate this cost. The function will be called four times (once for each cargo category). The function should be declared as:


```
function calcCost($cargoAmount, $pricePerUnit){}
```
- Print the spaceship identification number and the captain's name.
- For each of the four cargo categories, display/print the description, the sales amount, and the cost in column format using an **html table** (or css).
- Print the total paid to the company.
- If any form input is missing, `<my name>SpaceCargo.php` should be shown again.
- If any form input is missing, an error message should also be displayed, informing user the first missing form input.
- Use an external *.css file to format both .php files. Create your own *.css file. Do NOT use the provided *.css file in D2L.

Sample calculation (note that the formatting would be different on the actual web pages)

Example 1 Input:

Welcome to Jim Vandergriff's Space Travel Company	//your name
Please Enter Spaceship Identification	111
Please Enter the Captain's Name	John Smith
Please Enter Fuel Pod Sales	5
Please Enter Proton Cannon Ammunition Sales	2000
Please Enter Jetpack Sales	6

Please Enter Oxygen Sales	3
---------------------------	---

Output:

```

Jim Vandergriff's Space Travel Company Sales Statement //your name
Spaceship Identification Number 111 //captured input
Captain's Name: John Smith //captured input
*****
Cargo Sales Amount Cost
Fuel Pod 5 pods $627.50 //5 * 125.50
Cannon Ammunition 2000 rounds $344.00 //(2000/100) * 17.2
Jetpack 6 jetpacks $594.00 //6 * 99.00
Oxygen 3 tanks $151.20 //3 * 50.40
-----
Total Due $1716.70 //add 4 items above
=====

```

Example 2 Input:

```

Welcome to Jim Vandergriff's Space Travel Company
Please Enter Spaceship Identification or -999 to Terminate 113
Please Enter the Captain's Name Mary Jones
Please Enter Fuel Pod Sales 0
Please Enter Proton Cannon Ammunition Sales 3025
Please Enter Jetpack Sales 2
Please Enter Oxygen Sales 1

```

Output:

```

Jim Vandergriff's Space Travel Company Sales Statement //your name
Spaceship Identification Number 113 //captured input
Captain's Name: Mary Jones //captured input
*****
Cargo Sales Amount Cost
Fuel Pod 0 pods $0.00 //0 * 125.50
Cannon Ammunition 3025 rounds $520.30 //(3025/100) * 17.2
Jetpack 2 jetpacks $198.00 //2 * 99.00
Oxygen 1 tanks $50.40 //1 * 50.40
-----
Total Due $768.70 //add 4 items above
=====

```

Note: For this question, name your file <my name>SpaceCargo.php. For example, my file would be named: LeongLeeSpaceCargo.php. Change the page title to <my name> Space Cargo. Name the second file <my name>ProcessCargo.php with page title <my name> Process Cargo.

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 to 4 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_assignment02.pdf**.
2. All php files (and related css, htm, JavaScript files, if applicable). Zip your files into a single zip file (or rar file) **lastname_firstname_assignment02.zip**.

Please submit electronic copy (the above mentioned **two files**) 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**.