

CIS 122 Web Internet Technologies

Exam 2: PHP

100 points

Academic Honesty Policy:

Students are expected to uphold the school's standard of conduct relating to academic honesty. It is imperative that standards of academic integrity be upheld for the best interest of the student, college, community, and industry. Therefore, any instances of academic dishonesty (cheating) will result in an immediate submission of a failing course grade to the college records system. Violations resulting in a failing grade for the course will be forwarded to the Office of the Registrar.

Submission Requirements:

1. Transfer the entire **exam2** folder that contains your web pages, including subfolders, from your PC to the **cislinux** web server. Make sure it is located under your **public_html** folder and that all files have a **604** permission and all folders have a **701** permission.
2. Test your page by opening it in Chrome. The URL should look like this:
cislinux.hfcc.edu/~userid/exam2/temp.html
Just replace userid with your userid
3. Copy that URL and paste it to the comment section in the Moodle exam2 assignment
4. Upload the following files to our Moodle web site:
 - i. temp.html
 - ii. temp.css
 - iii. temp.php

Directions:

Part 0: Setup

1. Download the starter files and folders from Moodle:
 - a. Go to the link under Exam2 and download the set of files,
 - b. Unzip to the location on your PC you will be working from. Make sure you note the location you unzipped to, as you will later upload these updated files to **Moodle** and to the **cislinux** web server. You should now have a folder named **exam2**, with skeleton HTML, CSS, and PHP files and two subfolders (css and images) within it. These subfolders contain a skeleton temp.css file and a png image, respectively.

Part 1: HTML

2. Open the files **temp.html, temp.css, and temp.php** for editing. Put your name within comments on all three documents.
3. The HTML and PHP files should use **Weekly Average Temperature Calculator** in the <title> tag in the <head> area and in an <h1> tag in the <body> area.
4. The image should be inserted after the <h1> element, its alt attribute to read "Daily Weather".
5. **temp.html** will contain an HTML data entry form that will allow the user to enter 7 daily Fahrenheit temperatures (see screenshot below).
 - Use an 8 row by 2 column table inside the form to give it a structure.
 - Place the days of the week and the **number** input box in the first 7 rows.
 - Place the Submit button in the last table row.
6. Use the **number** HTML input type on the temperature fields and only allow numeric values between **-30 and 140** to be entered. Give each field a "required" attribute.
7. When the user clicks the "Submit" button, the user inputted data from **temp.html** will be sent to the PHP page **temp.php**
 - Select the values for the **name** attributes in the input fields (e.g., "monday", "tuesday", etc.) and make sure they are consistent with the PHP code.
 - Send the data using **either** the **post** or **get** method, just make sure the choice is consistent between the HTML and the PHP files.

Part 2: CSS

8. Link your **temp.html** to a new CSS stylesheet file named **temp.css** located in the **css** subfolder.
9. Add the CSS code to **temp.css** which will turn the input boxes and submit button to "palegreen".
10. Make the img element width to 20% of its normal width.
11. At this point test your HTML / CSS work. You can run this on your PC/Mac but the submit button will not work as intended.

Sample screenshot of temp.html output:

Weekly Average Temperature Calculator



Monday:	100
Tuesday:	100
Wednesday:	100
Thursday:	100
Friday:	100
Saturday:	100
Sunday:	100
Submit	

Part 3: PHP

12. The **temp.php** program will accept in the 7 pieces of data sent from the form. and **filter/sanitize the input** as was done in our homework assignments. For these 7 user inputs, use the PHP built-in functions `filter_input` and `substr` to ensure they are integer numbers and that no more than 4 characters are accepted in.
13. The **temp.php** program will then calculate the **average** Fahrenheit temperature based on the seven daily temperatures entered by the user.
14. **Round** the average temperature numeric output to 0 digit to the right of the decimal place. Display the average weekly temperature as shown in the screenshot below (with calculated average temperature).
15. Next, use a PHP decision (IF-ELSEIF-ELSEIF-ELSE) statements to display one of the following four message depending on the average temperature.

Message

That's hot.
That's warm.
That's chilly.
That's cold.

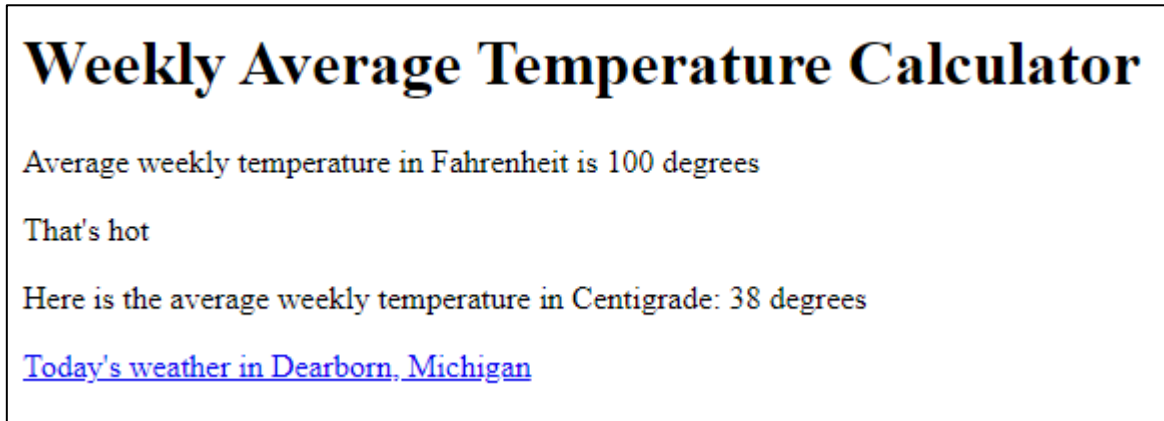
Condition

If average is 95 or higher
If average is 70 or higher
If average is 40 or higher
Anything else, i.e. less than 40

16. Next, use Google to find the formula to convert from Fahrenheit to Centigrade, then, coding that in PHP, convert this Fahrenheit average temperature to Centigrade units and display a line as shown in the screenshot below.
17. Insert a hyperlink to the URL below and make the text of the link read Today's weather in Dearborn, MI

<https://www.theweathernetwork.com/us/weather/michigan/dearborn>

The following is how the output to temp.php would look if the average temperature was 100:



Rubric

		Incomplete	Developing	Competent
Criteria 1: temp.html – Validates as HTML 5 at W3C.		0 pts Required code all missing or incorrect.	1 – 4 points Partially correct.	5 pts All correct.
Criteria 2: temp.html – Image tag correct.		0 pts Missing or all incorrect.	1 – 4 points Partially correct.	5 pts All correct.
Criteria 3: temp.html – HTML Input Form correct.		0 pts Missing or all incorrect.	1 – 29 points Partially correct.	30 pts All correct.
Criteria 4: temp.html – CSS code correct and working.		0 pts Missing or all incorrect.	1 – 4 points Partially correct.	5 pts All correct.
Criteria 5: temp.php – user input accepted in and filtered/sanitized correctly.		0 pts Missing or all incorrect.	1 – 4 points Partially correct.	5 pts All correct.
Criteria 6: temp.php – average temperatures calculated and displayed correctly.		0 pts Missing or all incorrect.	1 – 9 points Partially correct.	10 pts All correct.
Criteria 7: temp.php – Round function correct.		0 pts Missing or all incorrect.	1 – 4 points Partially correct.	5 pts All correct.
Criteria 8: temp.php – decision statement works and outputs correctly.		0 pts Missing or all incorrect.	1 – 29 points Partially correct.	30 pts All correct.
Criteria 9: temp.php – Hyperlink correct.		0 pts Missing or all incorrect.	1 – 4 points Partially correct.	5 pts All correct.