CS 1520 / CoE 1520: Programming Languages for Web Applications (Spring 2013)

Department of Computer Science, University of Pittsburgh

Assignment #2: PHP

Released: March 19th, 2013 **Due:** 11:59pm, Monday, April 1st, 2013

Goal

Gain familiarity with PHP.

Description

In this assignment, you will build a weather data Q&A service. You will rely on data from *The Weather Underground* (`http://www.wunderground.com/`) and provide answers to specific questions, given by the user.

Your program (`myweather.php`) should get user input via a simple HTML form that uses the GET method to send data to the web server. Your program should generate the HTML form if it is called for the first time and produce the requested results after clicking the submit button (i.e., the action for the form will be the same program, `myweather.php`).

Data

The weather data is available as comma-separated-value files (CSV), that can be downloaded from:
`http://www.wunderground.com/history/airport/PIT/2010/7/14/DailyHistory.html?format=1`
where:

- PIT is the airport code. Example codes are: PIT, IAD, SFO, ORD, PHL, JFK, LAX, MIA, and PHX.
- 2010 is the year
- 7 is the month (i.e., July)
- 14 is the day of the month
- format=1 means CSV.

You can view the same data in a nicely-formatted manner at:
`http://www.wunderground.com/history/airport/PIT/2010/7/14/DailyHistory.html`.

User Input

The HTML form that your program generates should collect the following user input:

- `airport`
- `year`
- `month` (numeric)
- `day` (optional)
- `function`, which should be one of the following: average, highest, lowest, conditions.

Note that:

- You should make sure your HTML form uses the above names (exactly) for the input fields.
- Users are not infallible and your program should gracefully handle cases where the wrong information is submitted (e.g., random string instead of a year) or the airport code is an invalid one.
- Your program should handle the different range of days for every month (e.g., 30 vs 31 vs 28 vs 29).
- The wunderground.com server "gracefully" handles out of range situations by computing the correct date. For example, "February 31, 2012" is converted to "March 2, 2012".

Functionality

Your program should support the following functionality:

1. Average Temperature (`function == 'average'`)
   This should compute (and print) the average temperature for every day of the month and also print the highest average temperature and the lowest average temperatures for the entire month, if no day is provided. If a day is provided by the user, then you should compute the average, highest, and lowest temperature for just that day.

   Note: to simplify your computations we will just treat every temperature reading independently and simply divide by the total number of readings to get the average (i.e., ignore the fact that there may be duplicate readings within the same one-hour period).

   Sample output:

   ```
   Station: PIT
   Query: average (year: 2012 / month: 7 / day: )
   July 1, 2012: 81F
   July 2, 2012: 78F
   July 3, 2012: 78F
   ....
   July 31, 2012: 77F

   Highest Average: 85F (July 6, 2012)
   Lowest Average: 71F (July 25, 2012)
   Lowest Average: 71F (July 28, 2012)
   ```

   Note: in cases of ties, you should print all matches (like the lowest average above).

2. Highest Temperature (`function == 'highest'`)
   This should compute (and print) the high temperature for every day of the month and also print the highest high temperature and the lowest high temperatures for the entire month, if no day is provided. If a day is provided by the user, then you should compute the highest temperature for just that day.

   The behavior is identical to the computation of the average (explained above), but instead of looking at averages you need to look at the high temperature per day (e.g., 94F for July 1, 2012).

3. Lowest Temperature (`function == 'lowest'`)
   This should compute (and print) the low temperature for every day of the month and also print the highest low temperature and the lowest low temperatures for the entire month, if no day is provided. If a day is provided by the user, then you should compute the lowest temperature for just that day.

   The behavior is identical to the computation of the average (explained above), but instead of looking at averages you need to look at the low temperature per day (e.g., 68F for July 1, 2012).

4. Conditions Frequency (`function == 'conditions'`)
   This should collect all the conditions data for the entire month (e.g., *Clear*, *Scattered Clouds*, *Mostly Cloudy*, etc) and report their frequency (i.e., percentage of occurrence), showing them in order of

decreasing frequency (i.e., most frequent conditions first), if no day is provided. If a day is provided by the user, then you should report the frequencies for just that day.

Note that, as with the averages, you should consider all readings as equal.

Sample output:

```
Station: PIT
Query: conditions
Month: July 2015

43.4%  Scattered Clouds
20.6%  Partly Cloudy
10.2%  Light Thunderstorms and Rain
 4.7%  Clear
....
```

What to submit

A PHP program that performs the tasks listed above, along with any additional libraries that you have developed (and are shared by your program). Name your program `myweather.php`. There is no standard naming scheme for the libraries.

Academic Honesty

The work in this assignment is to be done *independently*, by you and only you. Discussions with other students on the assignment should be limited to understanding the statement of the problem. **Cheating in any way, including giving your work to someone else, will result in an F for the course and a report to the appropriate University authority for further disciplinary action.**

How to submit your assignment

We will use a Web-based assignment submission interface. To submit your assignment:

- If you have more than one file to submit, prepare your assignment for uploading, by generating a single zip file with all the files.
- Go to the class web page `http://db.cs.pitt.edu/courses/cs1520/spring2013` and click the Submit button.
- Use your pittID as the username and the password you specified at the contact information form for authentication. There is a reminder service via email if you forgot your password. You must have already submitted your contact information, if you have not yet you need to do so now.
- Upload your assignment file to the appropriate assignment (from the drop-down list).
- Check (through the web interface) to verify what is the file size that has been uploaded and make sure it has been submitted in full. **It is your responsibility to make sure the assignment was properly submitted.**

You must submit your assignment before the due date (11:59pm, Monday, April 1st, 2013) to avoid getting any late penalty. The timestamp of the electronic submission will determine if you have met the deadline. There will be no late submissions allowed after 11:59pm, Wednesday, April 3rd, 2013.

[Last updated on March 19, 2013 at 1:11am EST]