

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT202-007-F2024/generic-module-5-multi-dimension-php-problems/grade/ob75>

Course: IT202-007-F2024

Assignment: [Generic] Module 5 Multi-Dimension PHP Problems

Student: Omar B. (ob75)

Submissions:

Submission Selection

1 Submission [submitted] 10/22/2024 9:05:01 PM

Instructions

^ COLLAPSE ^

Overview video: <https://youtu.be/lp568G93Noo>

Guide:

1. Make sure you're in the dev branch locally and `git pull origin dev` any pending changes.
2. Make a new branch per the recommended branch name below (`git checkout -b ...`).
3. Grab the template code from <https://gist.github.com/MattToegel/f7b0489fb0d8cee615d6626056ac5de2>
4. Create individual PHP files for each problem and save the files inside your `public_html` folder in a subfolder of your choice.
5. Move the unedited template files to GitHub.
 1. `git add .`
 2. `git commit -m "adding template files"`
 3. `git push origin branch_name` (see below)
 4. Create and open a pull request from the homework branch to main (leave it open until later steps).
6. Note: As you work, it's recommended to add/commit at least after each solution is done (i.e., 3+ times in this case).
 1. Make sure the files are saved before doing this.
7. Fill in the items in the worksheet below (save as often as necessary).
8. Once finished, export the worksheet.
9. Add the output file to any location of your choice in your repository folder (i.e., a `Module5` folder).
10. Check that git sees it via `git status`.
11. If everything is good, continue to submit.
 1. Track the file(s) via `git add`

1. Track the file(s) via `git add`.
 2. Commit the changes via `git commit` (don't forget the commit message).
 3. Push the changes to GitHub via `git push` (don't forget to refer to the proper branch).
 4. Create a pull request from the homework related branch to main (i.e., dev <- "homework branch").
 5. Open and complete the merge of the pull request (it should turn purple).
 6. Locally checkout dev and pull the latest changes (to prepare for future work).
12. Take the same output file and upload it to Canvas.

Branch name: M5-MD-PHP-Problems

Group

100%

Group: Problem 1

Tasks: 1

Points: 3

^ COLLAPSE ^

Task

100%

Group: Problem 1

Task #1: Problem 1 Evidence


Weight: ~100%

Points: ~3.00

^ COLLAPSE ^

Details:

Only make edits where the template code mentions.

Solution should add logic to create a new array with only name, color, and region (subset of the original data) 

Columns: 1

Sub-Task

100%

Group: Problem 1

Task #1: Problem 1 Evidence

Sub Task #1: Show the output from heroku dev (url must be visible)

Task Screenshots

Gallery Style: 2 Columns

4

2

1





Showing heroku URL and output of subset only containing name, color, and region in new array.

Caption(s) (required) ✓

Caption Hint: Describe/highlight what's being shown

Task URLs

URL #1

<https://it202-ob75-prod-62037cdcca6c.herokuapp.com/problem1.php>

<https://it202-ob75-prod-62037cdcca6c.herokuapp.com/problem1.php>

Sub-Task

Group: Problem 1

Task #1: Problem 1 Evidence

Sub Task #2: Show the code solution (ucid/date as comment must be present)

100%

Task Screenshots

Gallery Style: 2 Columns

4

2

1



Showing problem1 code with UCID and date from line 31-39 as relevant code.

Caption(s) (required) ✓

Caption Hint: Describe/highlight what's being shown

Task Response Prompt

Explain in concise steps how this logically works

Response:

For problem 1, a foreach loop was created to loop through the \$birds variable which contained all the arrays given and would go through each key, named as \$key, to get all the keys. Then I only added the specific keys and variables necessary to the subset array created for us by setting subset[] to each key which was 'name', 'color', 'region' and this was added to this array in array format by adding the brackets around them. The \$key['name'] for example grabbed the key value that matched with 'name' in all the arrays and added them to the subset array. This was done for the other 2 as well, giving the new subset array only the 3 necessary keys asked for.

End of Task: Problem 1

Task Status: 1/1

Group

100%

Group: Problem 2

Tasks: 1

Points: 3

⬆ COLLAPSE ⬆

Task

100%

Group: Problem 2

Task #1: Problem 2 Evidence

Weight: ~100%

Points: ~3.00

⬆ COLLAPSE ⬆

Details:

Only make edits where the template code mentions.

Solution should add logic to create a new array with original properties plus age and isClassic (extra data)

Columns: 1

Sub-Task

100%

Group: Problem 2

Task #1: Problem 2 Evidence

Sub Task #1: Show the output from heroku dev (url must be visible)

Task Screenshots

Gallery Style: 2 Columns

4

2

1



Showing Heroku URL and output with new array containing added values of isClassic and age.

Caption(s) (required) ✓

Caption Hint: Describe/highlight what's being shown

Task URLs

URL #1

<https://it202-ob75-prod-62037cdcca6c.herokuapp.com>

<https://it202-ob75->

prod-62037cdcca6c.herokuapp.com/problem2.php

Sub-Task

Group: Problem 2

Task #1: Problem 2 Evidence

Sub Task #2: Show the code solution (ucid/date as comment must be present)

100%

Task Screenshots

Gallery Style: 2 Columns

4

2

1

```

import numpy as np
import pandas as pd
import sys

# Create a sample DataFrame
df = pd.DataFrame({
    'Age': [25, 30, 35, 40, 45, 50, 55, 60, 65, 70],
    'Gender': ['M', 'F', 'M', 'F', 'M', 'F', 'M', 'F', 'M', 'F'],
    'Height': [1.75, 1.80, 1.85, 1.90, 1.95, 2.00, 2.05, 2.10, 2.15, 2.20],
    'Weight': [70, 75, 80, 85, 90, 95, 100, 105, 110, 115],
    'BloodPressure': [120, 130, 140, 150, 160, 170, 180, 190, 200, 210],
    'Cholesterol': [200, 220, 240, 260, 280, 300, 320, 340, 360, 380],
    'Glucose': [90, 100, 110, 120, 130, 140, 150, 160, 170, 180],
    'HemoglobinA1c': [5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5],
    'Diabetes': [0, 0, 0, 0, 0, 0, 1, 1, 1, 1],
    'Smoker': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Hypertension': [0, 0, 1, 1, 0, 0, 1, 1, 0, 0],
    'HeartRate': [70, 75, 80, 85, 90, 95, 100, 105, 110, 115],
    'SleepHours': [7, 8, 9, 10, 11, 12, 13, 14, 15, 16],
    'StressLevel': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'MentalHealth': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Anxiety': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Depression': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'SubstanceUse': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'FamilyHistory': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Lifestyle': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Environment': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'SocialSupport': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'WorkLifeBalance': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'FinancialStability': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'EducationLevel': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'MaritalStatus': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'ParentalStatus': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'PetOwnership': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Golfing': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Fishing': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Gardening': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Reading': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Traveling': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Volunteering': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'ReligiousBeliefs': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Spirituality': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Meditation': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Yoga': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Pilates': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Swimming': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Cycling': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Hiking': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Jogging': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Walking': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Sitting': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Standing': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Lying': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Sleeping': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Eating': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Drinking': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Smoking': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'UsingDrugs': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingSex': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'GettingMarried': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'GettingDivorced': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingChildren': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingChildren': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingPets': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingPets': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingFriends': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingFriends': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingFamily': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingFamily': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingCommunity': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingCommunity': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingCulture': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingCulture': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingTraditions': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingTraditions': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingValues': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingValues': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingBeliefs': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingBeliefs': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingFaith': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingFaith': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingHope': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingHope': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingLove': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingLove': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingKindness': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingKindness': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingPatience': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingPatience': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingTolerance': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingTolerance': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingRespect': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingRespect': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingGratitude': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingGratitude': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingForgiveness': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingForgiveness': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingCompassion': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingCompassion': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingEmpathy': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingEmpathy': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingSympathy': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingSympathy': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingPity': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingPity': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingGuilt': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingGuilt': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingShame': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingShame': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingEmbarrassment': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingEmbarrassment': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'HavingHesitation': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'LosingHesitation': [0, 1, 0
```

Showing problem2 code with UCID and date from line 33-51 as relevant code.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

⇒ Task Response Prompt

Explain in concise steps how this logically works

Response:

Firstly, `$currentYear` needed to be set to the current year since it was originally set at null so we set `$currentYear` to `date('Y')` which is the date format for only grabbing the current year. Then our new variable, `isClassic`, that is asked to be fed into the new array (`$processedCars[]`) is set to to boolean false here to then change it in our if statement later on if the car is classic. `foreach` is done to iterate through the given variable `$cars` to go through all the arrays getting all the keys. Our other new variable `$age` is set inside here to calculate the age of each car by subtracting the `currentYear` with the key value of the car's year doing `$key['year']`. An if statement is then placed to check if that age of the car is greater than or equal to the `classic_age` which would make the car 'isClassic'. If this was true, then we set `isClassic` to true, if not then it was set to false. We then feed all the keys to keep the original properties into a new temporary variable named `$newArray` and took all the `$key`. Our new variables `$isClassic` and `$age` is then fed into here as well to add the 2 new properties and finally our temporary variable is then put into the new array given to us, `$processedCars[]` and now shows us all our original properties along with each car's `$isClassic` boolean (true or false) and their `$age`.

End of Task 1

End of Group: Problem 2

Task Status: 1/1

Group

100%

Group: Problem 3

Tasks: 1

Points: 3

^ COLLAPSE ^

Task

100%

Group: Problem 3

Task #1: Problem 3 Evidence

Weight: ~100%

Points: ~3.00

^ COLLAPSE ^

Details:

Only make edits where the template code mentions.
Solution should add logic to join the arrays on userId
Requires at least 2 screenshots (code + output from heroku dev)
Live URL must be Herokue Prod



Columns: 1

Sub-Task

100%

Group: Problem 3

Task #1: Problem 3 Evidence

Sub Task #1: Show the output from heroku dev (url must be visible)

Task Screenshots

Gallery Style: 2 Columns

4

2

1



Showing heroku URL and output of joined array merged based on userId.

Showing bottom half of heroku output in case the first image does not show all of it.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task URLs

URL #1

[https://it202-ob75-](https://it202-ob75-162037cdcca6c.herokuapp.com/)

<https://it202-ob75-prod-62037cdcca6c.herokuapp.com/>

Sub-Task

Group: Problem 3

Task #1: Problem 3 Evidence

Sub Task #2: Show the code solution (ucid/date as comment must be present)

100%

Task Screenshots

Gallery Style: 2 Columns

4

2

1

```

function combine($users, $activities) {
    // Merge the $users and $activities variables to create one. Don't directly touch either array
    // and then merge them to form the array of arrays
    $temp = [];
    // Loop through $users
    foreach ($users as $user) {
        // Create an array to hold the user's activities
        $temp[$user['id']] = [];
    }
    // Loop through $activities
    foreach ($activities as $activity) {
        // Add the activity to the array of arrays
        $temp[$activity['userId']][] = $activity;
    }
    // Return the array of arrays
    return $temp;
}

// Example usage
$users = [
    ['id' => 1, 'name' => 'John'],
    ['id' => 2, 'name' => 'Jane'],
    ['id' => 3, 'name' => 'Bob'],
];

$activities = [
    ['userId' => 1, 'activity' => 'Reading'],
    ['userId' => 1, 'activity' => 'Golfing'],
    ['userId' => 2, 'activity' => 'Dancing'],
    ['userId' => 2, 'activity' => 'Cooking'],
    ['userId' => 3, 'activity' => 'Golfing'],
    ['userId' => 3, 'activity' => 'Reading'],
];

// Call the function
$result = combine($users, $activities);

// Print the result
print_r($result);

```

Showing code for problem3 with UCID and date and code starts from line 53-64 as relevant code.

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

For problem 3, I first created a temporary array named \$temp to add the userId key to for the \$users array side so that it can be matched against the activities array later on. foreach was done here to go through the list of \$users and find all the keys and was named as \$idKey. Once the \$idKey matched the key name 'userId' it was then added into to temp array as [idKey['userId']] and is now holding only our userId keys in the temp array for the users array end. Then another foreach was done on \$activities to go through all those keys (and named the same \$idKey). An if statement was placed to check if the temp isset with the variable idKey['userId'] it will then merge the idKeys of \$activities with the idKeys of \$users inside of the temp variable and finally set that to the \$joined[] which is the new array variable given to have all the combined \$users with their appropriate \$activities according to their keys and if they match. The output then shows each user with their corresponding activities based on the idKey values and if they match in both sections.

End of Task 1

End of Group: Problem 3

Task Status: 1/1

Group

Group: Reflection

Tasks: 3

Points: 1

100%

^ COLLAPSE ^

Task



Group: Reflection

Task #1: Reflect on your experience

Weight: ~33%

Points: ~0.33

^ COLLAPSE ^

Details:

Talk about any issues you had, how you resolved them, and anything you learned during this process.

Provide concrete details/examples. At least a few sentences.



≡ Task Response Prompt

Response:

The biggest issue in this assignment was moving arrays to another array but specifying only the properties needed to be moved into the new array. The way this was resolved was by creating for loops to go through the array and going through all the keys to check for the one we wanted. We then just specify the key that we wanted to move into the new array and added more properties if required. What was learned in this assignment was mostly how to access keys inside of an array and taking the values that you want to use and putting it into a new array to only display the key values that you want rather than having an extra clutter of data that is not needed.

End of Task 1

Task



Group: Reflection

Task #2: Include the pull request link for this branch

Weight: ~33%

Points: ~0.33

^ COLLAPSE ^

Details:

The correct link will end with /pull/ and a number.



🔗 Task URLs

URL #1

<https://github.com/OmarBarrera1/ob75-IT202-007/pull/19>

<https://github.com/OmarBarrera1/ob75-IT202-007>

End of Task 2

Task

100%

Group: Reflection

Task #3: Add Screenshot of Wakatime

Weight: ~33%

Points: ~0.33

COLLAPSE

Details:

Note: The duration of time isn't directly related to the grade, the goal is to just make sure time is being tracked



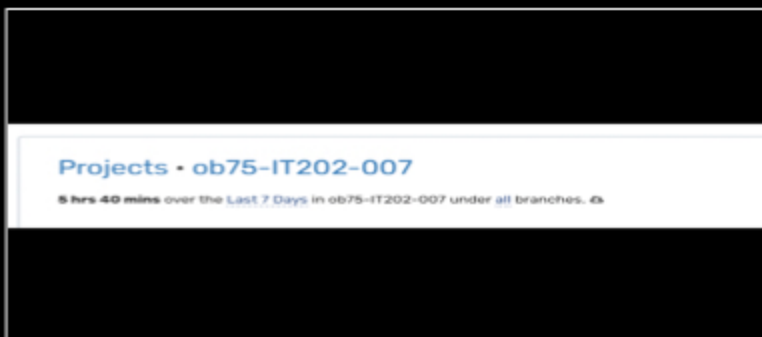
Task Screenshots

Gallery Style: 2 Columns

4

2

1



Showing wakatime top half, total time worked in the last 7 days.



Showing wakatime bottom half with time worked on each branch and files

End of Task 3

End of Group: Reflection

Task Status: 3/3

End of Assignment