PHP Practice

# Syntax Questions

### Q1. FizzBuzz

Write a function FizzBuzz which iterates the integers from 1 to 50. For multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”.

Expected Output

**Fizz Buzz Fizz Fizz Buzz Fizz FizzBuzz Fizz Buzz Fizz Fizz Buzz Fizz FizzBuzz Fizz Buzz Fizz Fizz Buzz Fizz FizzBuzz Fizz**

### Q2. Our old friend Mr. Triangle

Write a function printTriangle which takes in an integer **row** and returns a triangular pattern according to the row input.

|  |  |
| --- | --- |
| **Input - Row** | **Output** |
| **5** | **\***  **\* \***  **\* \* \***  **\* \* \* \***  **\* \* \* \* \*** |
| **7** | **\***  **\* \***  **\* \* \***  **\* \* \* \***  **\* \* \* \* \***  **\* \* \* \* \* \***  **\* \* \* \* \* \* \*** |

### Q3. PHPalindromes

Write a function isPalindrome that checks whether a passed string is a palindrome. If the input string is a palindrome, print “Palindrome!”, else print “Not a palindrome”. A palindrome is a word, phrase, or sequence that reads the same backwards as forward.

**Example 1:**

Input

$string = “Madam”;

Expected Output

Palindrome!

**Example 2:**

Input

$string = “Ellipsis”;

Expected Output

Not a Palindrome!

### Q4. The Oldies

Write a function oldest that takes in an associative array which contains the name and age of random people and returns the names of the oldest people.

**Example 1:**

Input

[‘Adam’ => 23, ‘Ben’ => 42, ‘Colin’ => 10, ‘Darren’ => 78, ‘Eric’ => 78, ‘George’ => 66]

Expected Output

Darren Eric

### Q5. The Old & Rich

Write a function oldestRichest that takes in **an array of associative array** that holds some folks’ data, and returns the name of the oldest person with the greatest salary.

\*if 2 or more people share the greatest age and salary, display the names of all with that age and salary.

**Example 1:**

Input

$array = [["name" => "Adam", "age" => 23, "Salary" => 200 ] ,

["name" => "Ben", "age" => 41, "Salary" => 105 ] ,

["name" => "Collin", "age" => 10, "Salary" => 78 ] ,

["name" => "Darren", "age" => 78, "Salary" => 2000 ] ,

["name" => "Eric", "age" => 78, "Salary" => 288 ] ,

["name" => "Farquar", "age" => 78, "Salary" => 2000]];

Expected Output

Darren Farquar

### Q6. Solve the Maze\*\*

You are given an M by N matrix consisting of booleans that represent a board. Each True boolean represents a wall, each False boolean represents a tile you can walk on.

Write a function minimumSteps that takes in a matrix, start coordinate and end coordinate that return the minimum number of steps required to reach the end coordinate from the start. If there is no possible path, then return **-1**. You can move up, left, down and right. You can’t move through walls. You cannot wrap around the edges of the board.

**Example 1:**

Input

$array = [[**False**, False, False, False],

[True, True, False, True],

[False, False, False, False],

[**False**, False, False, False]]

Start = **(3,0)** and end = **(0,0)**,the minimum number of steps required to reach the end is 7, since we would need to go through (1,2) because there is a wall everywhere else on the second row.

Output

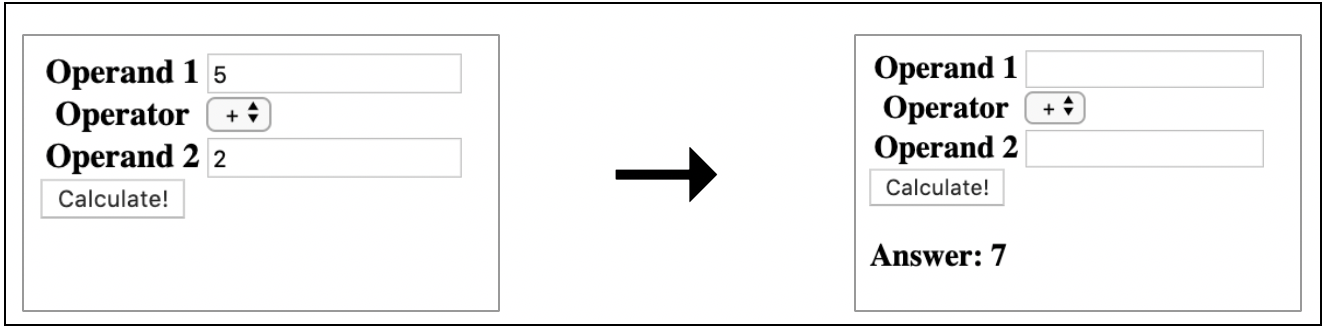
7

# Form Processing Questions

### Q7. Calculator

Create a calculator.php page that takes in 3 inputs (operand 1, 2 and operator) and displays the result of the numeric operation, as shown below.

The operators are +, - , \*, /



### 

### Q**8. Registration Page**

Create a registration.html page that takes in 2 inputs (Username and Password) and create registration.php to validate the username/password and display the message accordingly.

|  |  |
| --- | --- |
| **File** | **Display** |
| register.html |  |
| registration.php  Validation:   1. Username:    1. Must be at least 6 characters    2. Can only contain lowercase or uppercase letters 2. Password:    1. Must be at least 8 characters | Successful  Username: JackPhan  Password: Password    Fail  Username: jacky123 Password: Pass |

### Q9. Major Requirements

Create a select.php so that it displays a drop down menu with all the majors in School of Information System (SIS).

You are given a list of associative array $majors.

$majors = [

"SD" => "Software Development",

"DBS" => "Digital Business Solutioning",

"BA" => "Business Analytics",

"AI" => "Artificial Intelligence",

"CS" => 'Cyber Security',

"FT" => "Fintech"

];

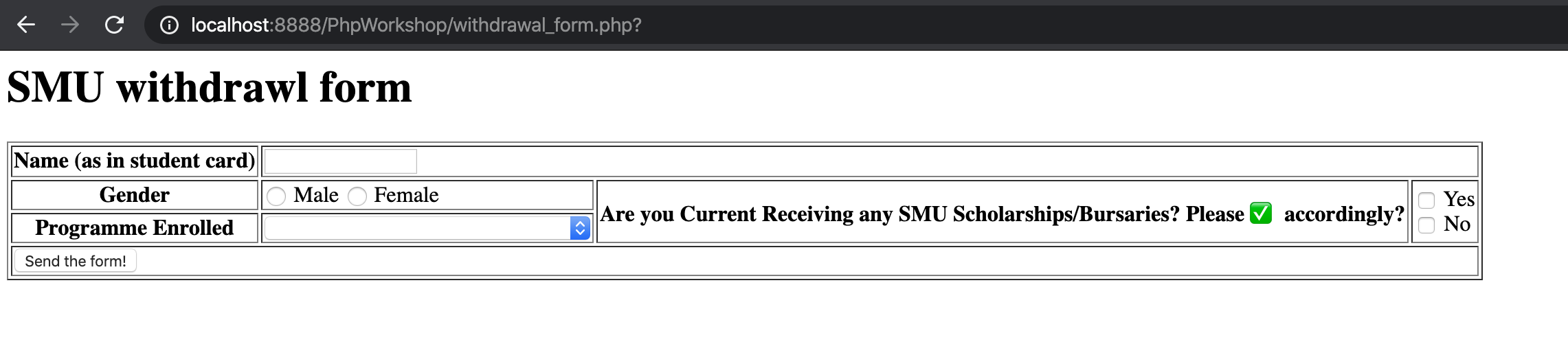
Create a module\_tracking.php to display all the modules requirements under each major in a table format.

You are given a list of associative array $modules (refer to resource file)

|  |  |
| --- | --- |
| **File** | **Display** |
| select.php |  |
| Module\_tracking.php  Example 1:  Major: Software Development | Expected Output: |

### Q10. SMU Withdrawal Form

The file withdrawal\_form.php has a form that submits a withdrawal form for validation.



The displayed choices for the respective fields are:

*Programme Enrolled:*

* Bachelor of Law
* Bachelor of Business
* Bachelor of Science (Information System)
* Bachelor of Economics
* Bachelor of Social Science

\*Student must fill in all fields

|  |  |
| --- | --- |
| Testcase 1 | All fields empty |
| User input |  |
| After Submission |  |
| Testcase 2 | Gender field blank |
| User input |  |
| After Submission |  |

# Classes and Object Questions

### Q11. Student Information Class

You are given the following Student class in student.php:

<?php

class Student {

private $name;

private $age;

private $hobbies;

function \_\_construct($name\_info, $age\_info, $hobbies\_info)

{

$this->name = $name\_info;

$this->age = $age\_info;

$this->$hobbies = $hobbies\_info;

}

#Create your own getters/setters

}

Your friend, Ernest, is 24. His hobby includes drinking.

Use this information to construct a Student object and append it to the $student\_information an array of student information.

<?php

$John\_info = new student("John",22,"Running");

$Jane\_info = new student("Jane",25,"Cooking");

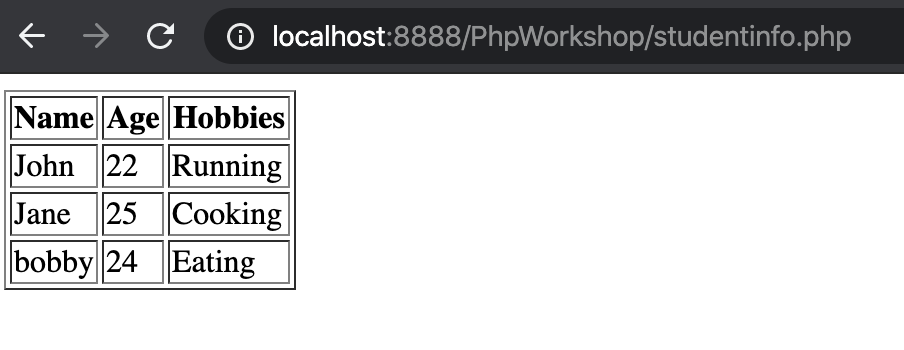
#Create Bobby Object

$student\_information = [$John\_info, $Jane\_info];

#Write your code here

?>

Display all the student info in a table:



### 

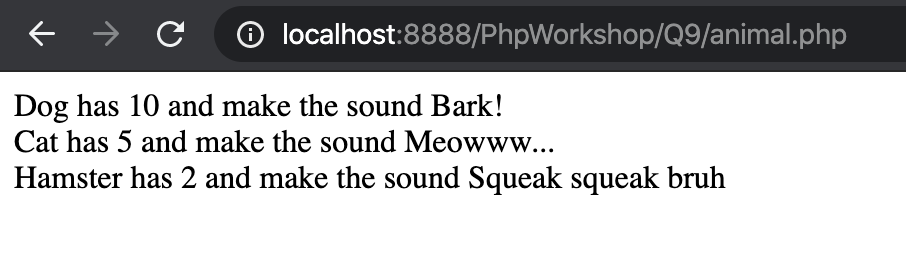
### Q12. Creating Animals

You are given a file called animal.php with un unimplemented class called Animal

Every animal should have the following attributes:

* Name
* Number of legs
* Noise

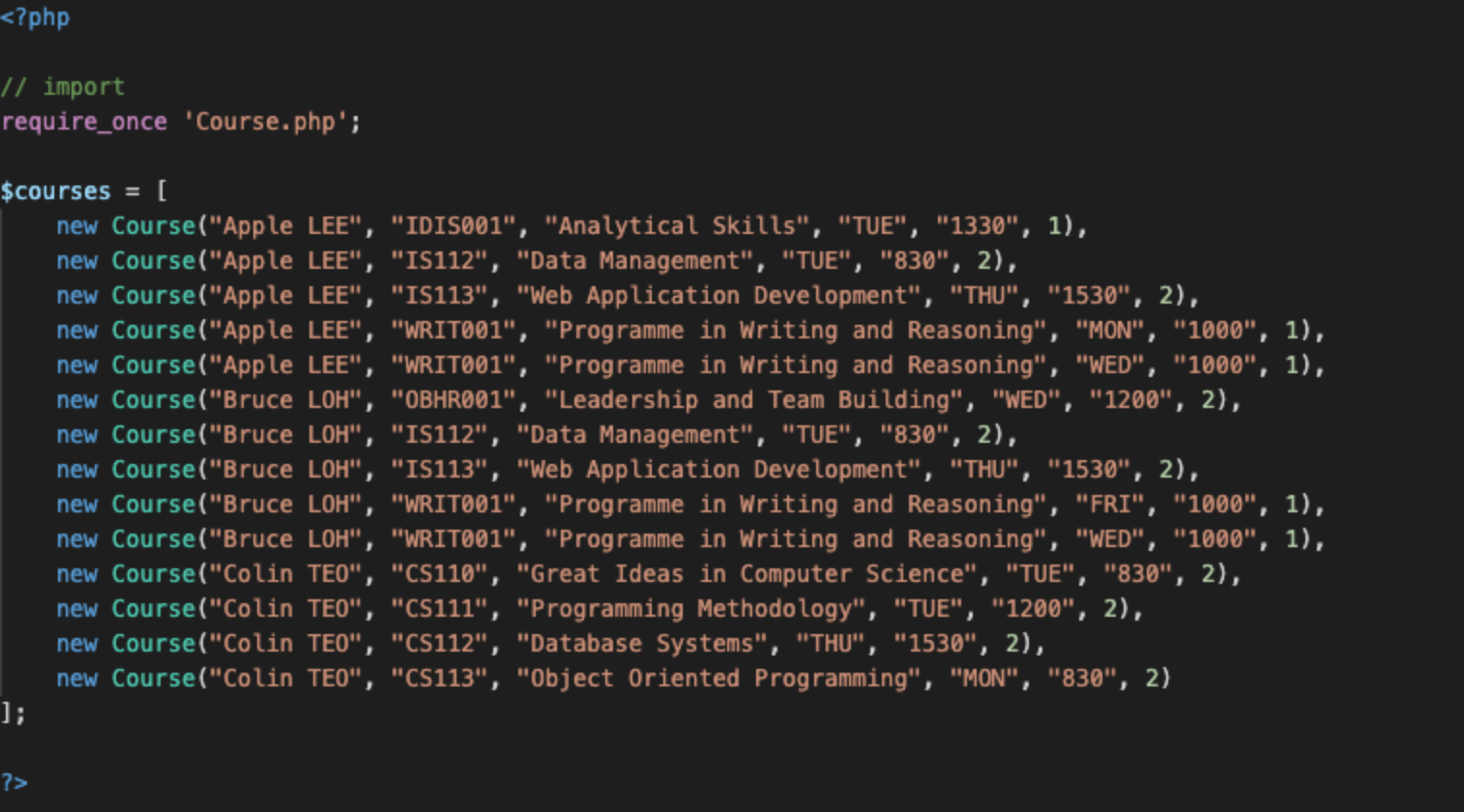
Try implementing the class such that the page renders the output shown below.



### Q13 Timetable

**You are given the Course class in course.php and $courses, an array of Course objects:**

****

****

|  |
| --- |
| **Question: You are a course coordinator for SMU and you wish to extract these students’ information and display it on a dynamic timetable webpage timetable.php such that Apple, Bruce and Colin can check their respective timetables online.**  **Your final output should look similar to this:**  **Example 1: Bruce Loh Selected**    **Example 2: Apple Lee Selected** |

# PDO (Database Interaction)

### Q14. Registration Page with Database

Refer to [Q8.Registration Page](#_heading=h.cm8esrcm76a5)

Only this time, after successful validation, insert the username and password into the **table** *user* under **database** *wadworkshop2021*.

We only need to store the username and password, so the userid is auto-incremented.



*Example of how the username and password is stored in database*

### 

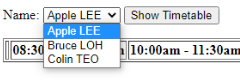
### Q15. Timetable with Database

Refer to [Q13 Timetable](#_heading=h.fa6rtf2894ln)

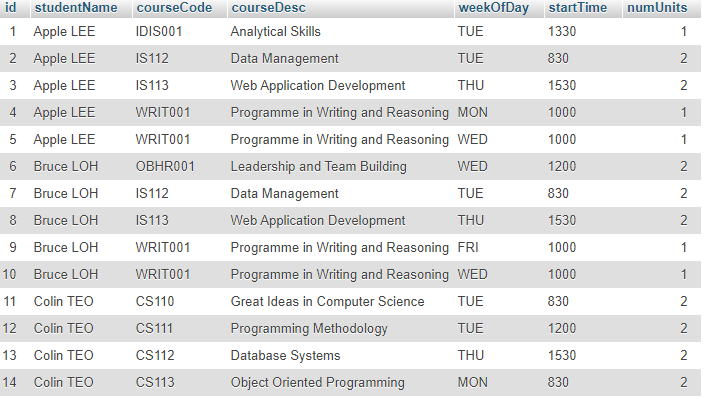
Only this time, all the timetable records are stored in the **table** *timetable* under **database** *wadworkshop2021.*

We can ignore the id as we wouldn’t be using it.

1. Pull the unique student names and display the dropdown list accordingly, from the data available inside the database.



2. And retrieve the required information to generate the timetable from the database and display it.



*Example of how timetable records is store in database*

**> All the best for finals!**