College of Engineering, Construction & Living Sciences
Bachelor of Information Technology
IN607: Introductory Application Development Concepts
Level 6, Credits 15

In-Class Activity: PHP Basics 1

Instructions

The purpose of this in-class activity to familiarise yourself with the **PHP** syntax as well develop your problem solving skills. The following 10 problems are commonly asked in coding interviews. You may come across one or two these when you apply for software development/engineering positions in the future.

Code Review

You must submit all program files via **GitHub Classroom**. Here is the URL to the repository you will use for your code review – https://classroom.github.com/a/P656imf2. Checkout from the **main** branch to the **01-in-class-activity** branch by running the command - **git checkout 01-in-class-activity**. This branch will be your development branch for this activity. Once you have completed this activity, create a pull request & assign the **GitHub** user **grayson-orr** to a reviewer. **Do not** merge your own pull request.

Problem 1:

Declare two variables called **name** & **age** with the values John & 55. Use the two variables to display the expected output.

```
<?php
  // Write your solution here

  // Expected output:
  // Hello my name is John & I am 55 years old.
?>
```

Problem 2:

Calculate the **sum** of the given **integers** & display the expected output.

```
<?php
```

```
  $x = 1957452;
  $y = 2975635;

  // Write your solution here

  // Expected output:
  // The sum of 1957452 & 2975635 is 4933087
?>
```

Problem 3:

Calculate the **average** of the given **array** of **doubles** & display the expected output.

```
<?php
  $numbers = array(45.3, 67.5, -45.6, 20.34, -33.0, 45.6);

  // Write your solution here

  // Expected output:
  // Average: 16.69
?>
```

Problem 4:

Write a function called **fizzBuzz** which accepts an **integer num**. If **num** is a multiple of three, return **Fizz**, if **num** is a multiple of five, return **Buzz** & if **num** is a multiple of three & five, return **FizzBuzz**. Call the **fizzBuzz** function in the for loop to display the expected output.

```
<?php
  // Write your fizzBuzz function here

  for ($i = 1; $i <= 15; $i += 2) {
    // Write your solution here
  }

  // Expected output:
  // 1
  // Fizz
  // Buzz
  // 7
  // Fizz
  // 11
  // 13
  // FizzBuzz
?>
```

Problem 5:

You have been given an **array** of **floats** or **doubles**. Display **only** the odd numbers in the **array**. Sort from lowest to highest.

```
<?php
  $numbers = array(21, 19, 68, 55, 42, 12);

  // Write your solution here
```

```
  // Expected output:
  // 19
  // 21
  // 55
?>
```

Problem 6:

Write a function called **is_anagram** which accepts two parameters called **string_one** & **string_two**. In the function block, write some code that checks whether or not **string_one** & **string_two** are an anagram. An anagram is a word or phrase that made by arranging the letters of another word or phrase in a different order. If you are still unsure what an anagram is, here is an example:

```
Input: is_anagram("elvis", "lives");
Output: true

Input: is_anagram("cat", "sat");
Output : false
```

Call the **is_anagram** function to display the expected output.

```
<?php
  // Write your solution here

  // Expected output:
  // true
  // false
?>
```

Problem 7:

Write a function called **convert** which accepts two parameters called **hours** & **minutes**. In the function block, write some code that converts both **hours** & **minutes** to seconds, then adds them together.

```
<?php
  // Write your solution here

  convert(1, 3);

  // Expected output:
  // 3780
?>
```

Problem 8:

Write a function called **palindrome** which accepts a single parameter called **string**. In the function block, determine whether or not **string** is a palindrome. The function should return a **boolean**.

```
<?php
  // Write your solution here

  palindrome("A man, a plan, a canal - Panama");
  palindrome("Hello, World!");

  // Expected output:
  // true
  // false
?>
```

Problem 9:

Write a function called **is_five_letters** which accepts an **array** of **strings**. In the function block, return all words that are exactly **five** letters.

```
<?php
  // Write your solution here

  is_five_letters(["car", "bike", "truck"]);

  // Expected output:
  // ["truck"]
?>
```

Problem 10:

Write a function that accepts an **integer**. If the **integer** is prime, return **true**, otherwise return **false**.

```
<?php
  // Write your solution here

  is_prime(11);
  is_prime(18);

  // Expected output:
  // true
  // false
?>
```