Tangent Technical Test

Q. No. 1: In object-orientated programming, what is the difference between a class and an object?

Answer:

Classes and **objects** are two main aspects of object-oriented programming. A class is a self-contained, independent set of variables, methods and behaviours which work together to perform specific logical tasks, while objects are individual instances of the class.

Class:

A class is like a **blueprint** for an object, which includes local methods and data. A class can be declared with the **class** keyword followed by the **name** of the class and curly braces e,g.

```
<?php
class Vehicle
{
    // Declare  properties
    public $make = '';
    private $transmission = '';

    // Constructor
    public function __construct($transmission) {
        echo 'Created an object of MyClass';
        $this->transmission = $transmission ;
    }


    // Method to get the make
    public function getMake(){
        return 'The make is:' .$this->make;
    }

    // Method to get the make
    public function getTransmission(){
        return 'The transmission is:' .$this->transmission;
    }
}
```

Objects:

An object is an **instance** of the class. Once a **class** has been created, you can instantiate as many **objects** as you want which will have the properties and behaviour of that class. An object can be created with the **new** keyword followed by the class name. e.g.

```
    // Making object of the classs
    $car = new Vehicle('auto');
    $car->make = 'Ford';
    // Accessing the attributes and methods of the class
    echo $car->make;
    echo $car->getTransmission();
```

Q. No. 2: Write a PHP program that is responsible for filling a bath. You can define any API you like to control the bath.

Answer:

```
        <?php

            class ProcessBath
            {
                private bool $tapsOpen = false;
                private float $waterTemperature = 25.0; // Default
temperature in Celsius
                private float $waterVolume = 0.0; // Volume in liters

                public function setWaterTapsOpen(bool $val)
                {
                    $this->tapsOpen = $val;
                }

                public function openTaps()
                {
                    if (!$this->tapsOpen) {
                        $this->tapsOpen = true;
                        $this->fillBathAsync();
                    } else {
                        echo "Taps are open already. Water is over-
flowing.\n<br>";
                        $this->waterVolume = 60.0;
                    }
                }

                public function closeTaps()
                {
                    if ($this->tapsOpen) {
                        $this->tapsOpen = false;
                        echo "Taps closed. Your bath is ready.\n<br>";
                    } else {
                        echo "Taps are already closed. Bath is already
ready or was not filled.\n<br>";
                    }
                }

                public function isBathReady(): bool
                {
                    return !$this->tapsOpen;
                }
```

```
                private function fillBathAsync()
                {
                    if(!$this->isBathReady()){
                        // Simulate asynchronous bath filling
                        for ($i = 0; $i < 10; $i++) {
                            $this->waterVolume += 5.0; // Simulating 5
liters of water added per iteration
                            usleep(250000); // Simulating a delay of 0.25
seconds (250,000 microseconds)
                            $this->updateTemperature();
                            echo "Filling bath: {$this->waterVolume}
liters, Temperature: {$this->waterTemperature}°C\n<br>";
                        }
                    }
                }

                private function updateTemperature()
                {
                    // Simulate temperature change during bath filling
                    $this->waterTemperature += 0.5; // Simulating a
temperature increase of 0.5°C per iteration
                }

                public function getWaterTemperature(): float
                {
                    // Accessing the water temperature after filling the
bath
                    return $this->waterTemperature;
                }

                public function getWaterVolume(): float
                {
                    // Accessing the water temperature after filling the
bath
                    return $this->waterVolume;
                }
            }

            // Usage with dafault behavior
            $bath = new ProcessBath();

            // Open the taps
            $bath->openTaps();

            // Simulate time passing or other activities

            // Close the taps
            $bath->closeTaps();

            // Check if the bath is ready
            if ($bath->isBathReady()) {
                echo "Bath is ready! Enjoy your relaxing bath. Water
temperature: {$bath->getWaterTemperature()}°C, Volume: {$bath-
>getWaterVolume()} liters.\n<br>";
            } else {
                echo "Bath is not ready. Please make sure to close the
taps.\n<br>";
            }
            echo '<hr>';
            // Usage with custom input behavior
            $bath = new ProcessBath();
```

```
            $setWaterTapsOpen = true; // User input
            $bath->setWaterTapsOpen($setWaterTapsOpen);

            // Open the taps
            $bath->openTaps();

            // Simulate time passing or other activities

            if($setWaterTapsOpen) {
                // Close the taps
                $bath->closeTaps();
            }

            // Check if the bath is ready
            if ($bath->isBathReady()) {
                echo "Bath is ready! Enjoy your relaxing bath. Water
temperature: {$bath->getWaterTemperature()}°C, Volume: {$bath-
>getWaterVolume()} liters.\n<br>";
            } else {
                echo "Bath is not ready. Please make sure to close the
taps.\n<br>";
            }
```

Q. No. 3: Write a short PHP function that reverses a string.

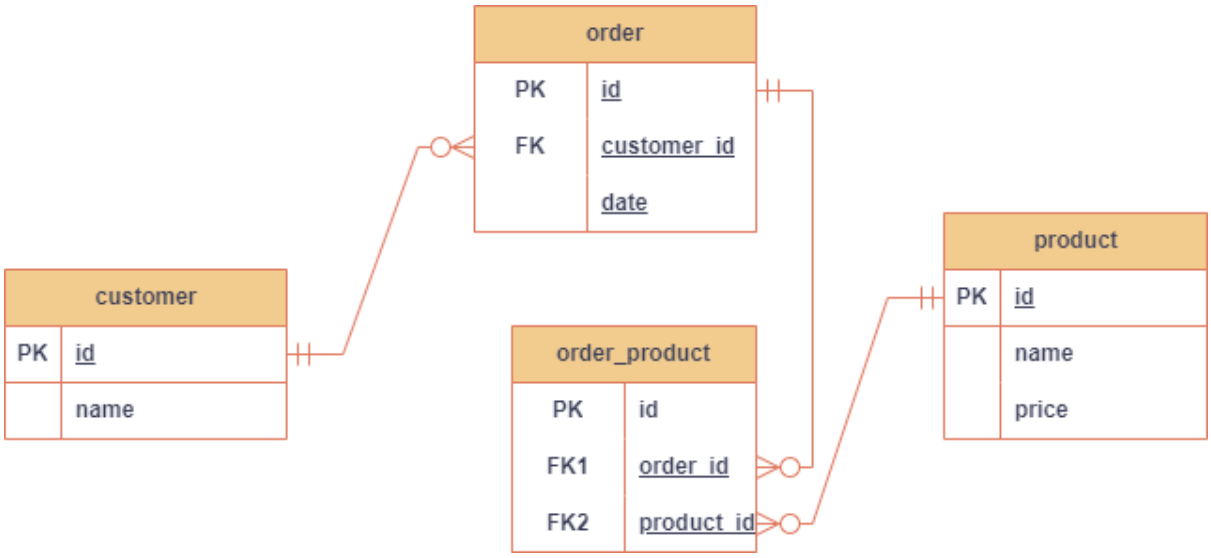
Answer:

```
        <?php
            function stringReversal($string) {
                // if the string has only one character or a empty string
                if ($string === '' || strlen($string) === 1) {
                    return $string;
                } else {
                    // Recursive case
                    return substr($string, -1) .
stringReversal(substr($string, 0, -1));
                }
            }

            $string = "A random string";
            $reversedString = stringReversal($string);

            echo "Original String: $string <br> Reversed String:
$reversedString";

            // Output
            // Original String: A random string
            // Reversed String: gnirts modnar A
```

Q. No. 4: Create a simple entity relationship diagram (ERD) that depicts the relationship between the following tables.

customer	order	order_product	product
id	id	id	id
name	customer_id	order_id	name
	date	product_id	price

Answer:



Q. No. 5: Finally, write an SQL query that returns the top 5 customers every day for the last month based on the database tables outlined in Question 4.

Answer:

Also, attached the database schema and data set to test the query.

```
        SELECT
            date,
            customer_id,
            customer_name,
            total
        FROM (
            SELECT
                o.date,
                c.id AS customer_id,
                c.name AS customer_name,
                SUM(p.price) AS total,
                ROW_NUMBER() OVER (PARTITION BY o.date ORDER BY
SUM(p.price) DESC) AS r_num
            FROM
                `order` o
            LEFT JOIN
                customer c ON o.customer_id = c.id
            LEFT JOIN
                order_product op ON o.id = op.order_id
            LEFT JOIN
```

```
                product p ON op.product_id = p.id
            WHERE
                o.date BETWEEN (select last_day(curdate() - interval 2
month) + interval 1 day) AND (select last_day(curdate() - interval 1
month))
            GROUP BY
                o.date,
                c.id
            ORDER BY `o`.`date` DESC, total DESC
        ) sub_data
        WHERE
            r_num <= 5;
```