**SOLUTION TO TECHNICAL TEST**

1. In object-orientated programming, what is the difference between a class and an object

**Answer**

• Class describes the contents of the object. A class is a template/entity that determines how an object will be created and what the object will contain. In other words, it is a blueprint or a set of instruction to build a specific type of object.

• An object is an instance of a class and with its own set of property values. It has its own state (property values) and can perform actions (invoke methods).

Example:

<?php

class Dog {

/\* Properties \*/

private $size;

private $colour;

private $age;

/\* Constructor method \*/

public function \_\_construct($size, $colour, $age) {

$this->size = $size;

$this->colour = $colour;

$this->age = $age;

}

/\* Method \*/

public function getInfo() {

echo ‘The size is: ’ . $this->size . ‘, The colour is: ’ . $this->colour. ‘ and the Age is: ’ . $this->age ;

}

}

// Example usage

// Creating objects (instances) of the Dog class

$dog1 = new Dog(‘Big’, ‘Brown’, ‘5’);

$dog2 = new Dog(‘Small’, ‘white’, ‘3’);

// Let’s Access the object properties and invoking methods

echo $dog1->size; // Output: Big

$dog2->getInfo(); // Output: The size is: Small, The colour is: white and the Age is: 3

?>

1. 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 Bath {

private $waterLevel = 0;

private $capacity;

private $tapsOpen = false;

public function \_\_construct($capacity) {

$this->capacity = $capacity;

}

public function openTaps() {

$this->tapsOpen = true;

echo "Taps are open. ";

}

public function closeTaps() {

$this->tapsOpen = false;

echo "Taps are closed. ";

}

public function fill($amount) {

if (!$this->tapsOpen) {

echo "Taps are closed. Cannot fill the bath. ";

return;

}

$this->waterLevel += $amount;

if ($this->waterLevel > $this->capacity) {

$this->waterLevel = $this->capacity;

echo "Bath is now full. ";

} else {

echo "Adding $amount liters of water. ";

}

}

public function drain($amount) {

if (!$this->tapsOpen) {

echo "Taps are closed. Cannot drain the bath. ";

return;

}

$this->waterLevel -= $amount;

if ($this->waterLevel < 0) {

$this->waterLevel = 0;

echo "Bath is now empty. ";

} else {

echo "Draining $amount liters of water. ";

}

}

}

// Example usage

$bath = new Bath(50); // Bath capacity is 50 liters

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

$bath->fill(10); // Fill the bath

$bath->drain(5); // Drain the bath

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

echo "Bath is ready!\n";

?>

1. Write a short PHP function that reverses a string

**Answer**

<?php

function reverseString($inputString) {

$stringLength = strlen($inputString);

$reversedString = '';

for ($i = $stringLength - 1; $i >= 0; $i--) {

$reversedString .= $inputString[$i];

}

return $reversedString;

}

// Example usage

$result = reverseString("Samson Ajakaye");

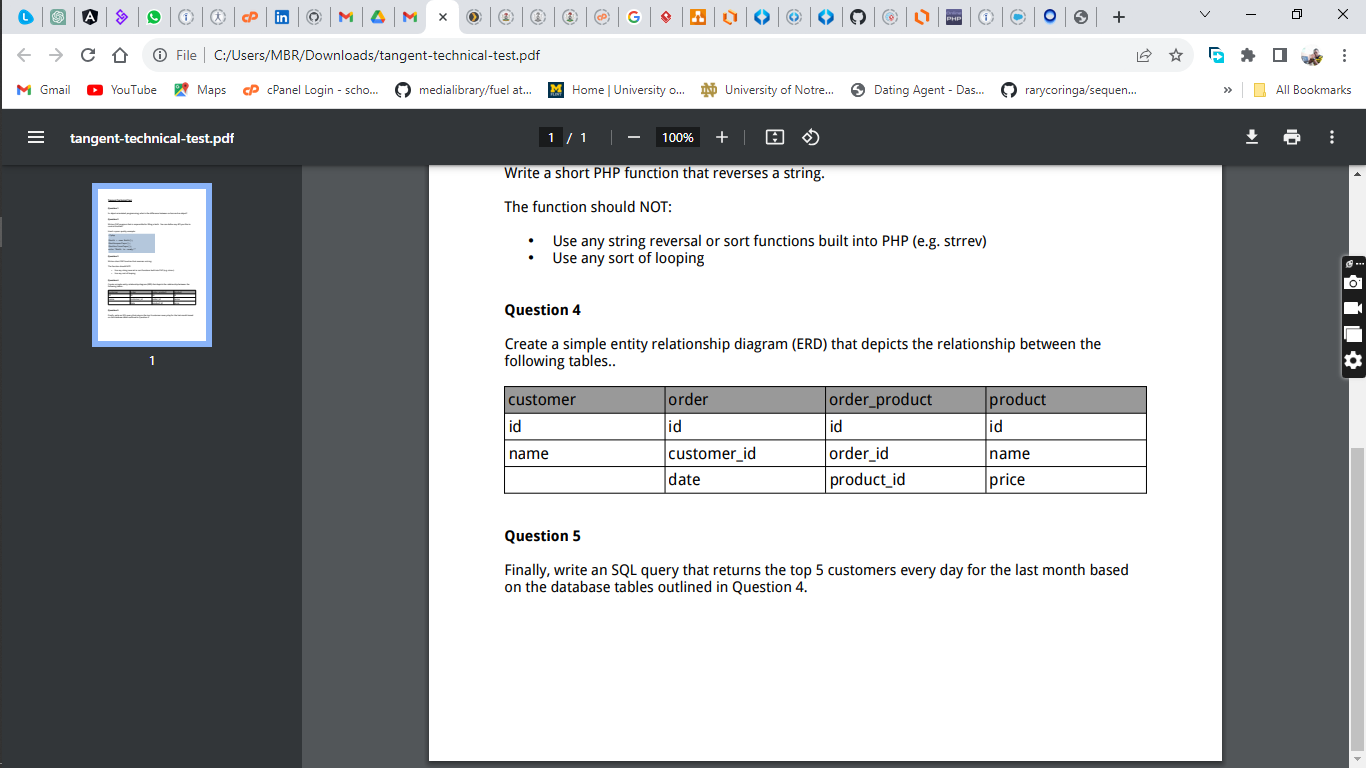
print\_r($result);

// Out put

eyakajA nosmaS

?>

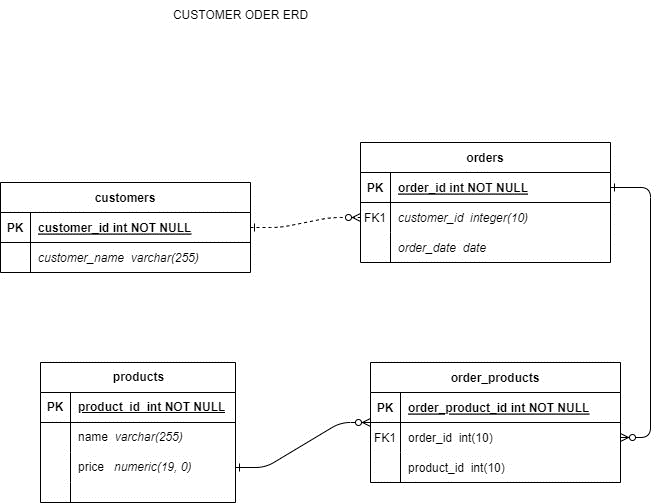
1. Create a simple entity relationship diagram (ERD) that depicts the relationship between the following tables.
2. Customer
3. order
4. order\_product
5. product



**Answer**

Relationships:

1. One order is associated with one customer (Many-to-One relationship between Order and Customer).
2. One customer can place multiple orders (One-to-Many relationship between Customer and Order).
3. Each order can have multiple order products (One-to-Many relationship between Order and Order\_Product).
4. Each product can be associated with multiple order products (One-to-Many relationship between Product and Order\_Product).



1. 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**

SELECT

c.customer\_id,

c.customer\_name,

DATE(o.order\_date) AS orderDate

FROM

`orders` o

JOIN

customers c ON o.customer\_id = c.customer\_id

JOIN

order\_products op ON o.order\_id = op.order\_id

JOIN

product p ON op.product\_id = p.product\_id

WHERE

o.order\_date >= CURDATE() - INTERVAL 1 MONTH

GROUP BY

orderDate,

c.customer\_id,

c.customer\_name

ORDER BY

orderDate DESC

LIMIT 5;