EAST WEST UNIVERSITY

CSE479
Web Programming

Nishat Tasnim Niloy

Lecturer

Department of Computer Science and Engineering

Faculty of Science and Engineering

Final Exam Exercises

(PHP, MySQL, NoSQL, XML, JSON and Security)

Example 1

Read the following Abstract Class-

```php
abstract class TopWear {
    protected $collar;
    protected $sleeve;
    protected $hood;

    abstract public function wear();
    abstract public function washInstructions();

    public function displayDetails() {
        echo "Collar: " . ($this->collar ? "Yes" : "No") . "\n";
        echo "Sleeve: " . $this->sleeve . "\n";
        echo "Hood: " . ($this->hood ? "Yes" : "No") . "\n";
    }
}
```

Now, create three child classes named- “Shirt”, “TankTop”, and “Hoody” that extends this TopWear class and implement the wear() and washInstructions() methods. Create a main class to execute all these methods from each child class. Do not forget to create constructors when it is necessary.

Example 2

Create a MySQL table named “Book” where, it contains Unique ID, title, author, genre, and edition. Now insert the following data-

ID	Title	Author	Genre	Price
1	To Kill a Mockingbird	Harper Lee	Fiction	7.99
2	1984	George Orwell	Dystopian	8.99
3	The Catcher in the Rye	J.D. Salinger	Fiction	6.99

Example 3

You want to store the information of books in a non-rational database. Each book must contain an ID, title, Author, and price. One book may contain multiple genres. Some of those may contain the Edition number. Now create a collection named “E_Book” under a database named “My_DB” using MongoDB based on this Information. You need to **insert** the data into your collection-

ID	Title	Author	Genre	Price	Edition
1	To Kill a Mockingbird	Harper Lee	Fiction	7.99	
2	1984	George Orwell	Dystopian, Tragedy	8.99	3
3	The Catcher in the Rye	J.D. Salinger	Fiction	6.99	

Example 4

Suppose a client needs a database design for his blog/website and website has the following requirements-

- Every post has the unique title, description and url.
- Every post can have one or more tags.
- Every post has the name of its publisher and total number of likes.
- Every post has comments given by users along with their name, message, data-time and likes.
- On each post, there can be zero or more comments.

Now, prepare a collection in MongoDB based on these requirements.

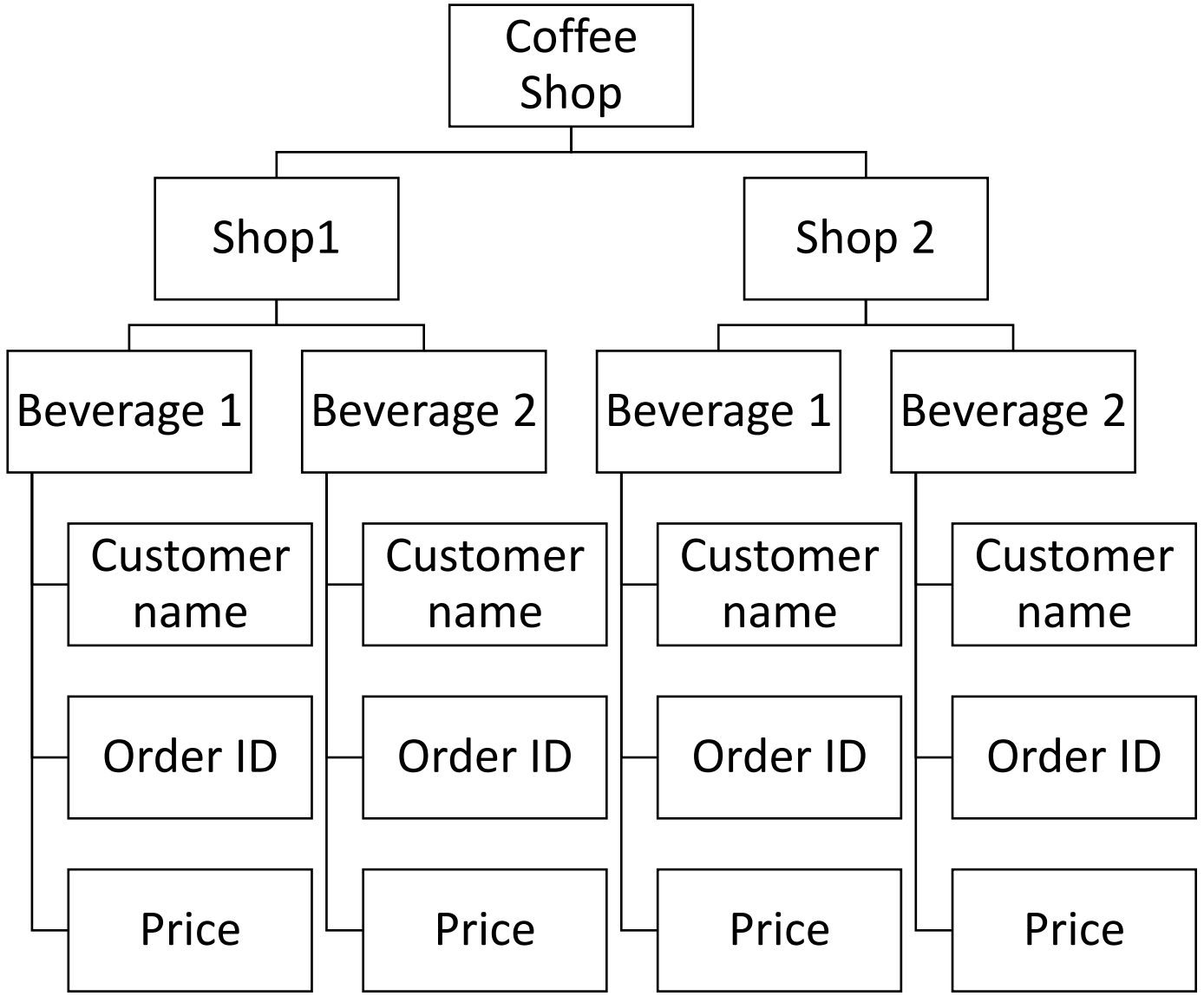
Example 5

Write a three-level nested XML and JSON document to represent the coffee shop's structure, following these guidelines:

- **Root Element**: "coffee Shop": The root object that contains the entire JSON document.
- **First Level:** "shop": Represents different coffee shops in the chain. Each shop is an object and contains a "location" attribute. You need to create two shops with "location" values for "New York" and "Los Angeles".
- **Second Level:** "beverages": Represents individual beverages served in each coffee shop. Each beverage is an object with a "code" attribute. The "code" values can be typical, such as "NY101" for a New York beverage or "LA101" for a Los Angeles beverage.
- **Third Level:** "name" and "orders": Represent details about each beverage and the customers who ordered them. The "orders" array contains order objects, and each order object has "customer name" and "order ID" attributes to provide more information about the orders. It would be enough if you created one order for each beverage.

Example 6

- Write a three-level nested XML and JSON document to represent the coffee shop's structure, following the tree structure:



Example 7

- Find the output-

```
<?php
class Member
{
public $username = "";
private $loggedIn = false;
public function login() {
$this->loggedIn = true;
}
public function logout() {
$this->loggedIn = false;
}
public function isLoggedIn() {
return $this->loggedIn;
}
}

$member = new Member();

$member->username = "Fred";

echo $member->username . " is " . (  $member->isLoggedIn()
    ? "logged in" : "logged out" ) . "<br>";

$member->login();

echo $member->username . " is " . ( $member->isLoggedIn()
    ? "logged in" : "logged out" ) . "<br>";

$member->logout();

echo $member->username . " is " . ( $member->isLoggedIn()
    ? "logged in" : "logged out" ) . "<br>";

?>
```

Example 8

- Find the output-

```
<?php
class Factorial
{
private $result = 1;
private $number;
function __construct($number)
{
$this->number = $number;
        for($i=2; $i<=$number; $i++)
            {
                $this->result*=$i;
            }
        echo "__construct() executed. ";
}
public function showResult()
{
echo "Factorial of {$this->number} is {$this->result}. ";
}
}
$fact = new Factorial(5);
$fact->showResult();
public function __destruct()
{
echo "I'm about to disappear - bye bye!";
}
?>
```

Example 9

- Find the output-

```
<?php
 class MyClass
{  public $prop1 = "I'm a class property!";
   public function __construct()
 {     echo 'The class "', __CLASS__, '" was initiated!<br />';
 }
 public function __destruct()
 {    echo 'The class "', __CLASS__, '" was destroyed.<br />';
 }
      public function __toString()
 {   echo "Using the toString method: ";
    return $this->getProperty();
 }
 public function setProperty($newval)
 {     $this->prop1 = $newval;   }
public function getProperty()
 {     return $this->prop1 . "<br />“;    }
}
class MyOtherClass extends MyClass
{  public function __construct()
 {   parent::__construct(); // Call the parent class's constructor
     echo "A new constructor in " . __CLASS__ . ".<br />";
 }
 public function newMethod()
 {  echo "From a new method in " . __CLASS__ . ".<br />";
 }
}
$newobj = new MyOtherClass;
echo $newobj->newMethod();
echo $newobj->getProperty();
?>
```

Example 10

- Find the output-

```
<?php
abstract class Shape {
   private $x = 0;    private $y = 0;
   public abstract function area();
}
class Rectangle extends Shape {
 function __construct($x, $y) {
       $this->x = $x;       $this->y = $y;
   }
  function area() {
    return $this->x * $this->y;
   }
}
class Square extends Shape {
   function __construct($x) {
      $this->x = $x;    }
   function area() {
      return $this->x * $this->x;    }
}
class Circle extends Shape {
   function __construct($x) {
      $this->x = $x;
   }
   function area() {
        return 3.1416 * $this->x * $this->x;
   }
}
$shapes = new Square(5);
echo "Square Area: " . $shapes->area() . "</br>";
$shapes = new Rectangle(12, 4);
echo "Rectangle Area: " .  $shapes->area() . "</br>";
$shapes = new Circle(2);
echo "Circle Area: " .  $shapes->area() . "</br>";
?>
```

Example 11

Suppose, you are going to develop a SNS application. To preserve the clients' and server's confidentiality, you need to design the usages of session, cookies and token. Explain, where and how will you manage their application in your system?