Advanced PHP Excercises

1)Define Object-Oriented Programming (OOP) and its four main principles: Encapsulation, Inheritance, Polymorphism, and Abstraction.

Object-Oriented Programming (OOP) is a way of writing organing code around objects.

objects represent real-world things or ideas. Each object has its own data attributes and methods,functions. OOP helps make code easier to understand, reuse, and maintain by how we think about things in real life

1. Encapsulation
   * Hides the inner details . like a remote control: we see its buttons interface, but you don’t need to know how it works inside.
2. Inheritance
   * Lets make new objects child classes based on existing parent classes.
   * The new object automatically gets the parent’s properties and methods—just like a child inherits from their parents. This reduces repeating code.
3. Polymorphism
   * Means many forms. we can use the same function or method name with different behaviors based on the object using it.
4. Abstraction
   * Hides unnecessary details and shows only the important stuff.

2)Create a simple class in PHP that demonstrates encapsulation by using private and public properties and methods.

<?php

class Person {

private $name;

private $age;

public function setName($name) {

$this->name = $name;

}

public function getName() {

return $this->name;

}

public function setAge($age) {

if ($age > 0) {

$this->age = $age;

}

}

public function getAge() {

return $this->age;

}

}

$person = new Person();

$person->setName("Rahul");

$person->setAge(25);

echo $person->getName();

echo $person->getAge();

?>

3) Explain the structure of a class in PHP, including properties and methods.

A class in PHP is a blueprint that defines the structure and data and actions of objects we create

* Variables inside a class. They store information about an object also called attributes.

Functions inside a class. They define what the class can do actions.

Example: For a Car class, methods might be startEngine(), displayInfo()

4) Write a PHP script to create a class representing a "Car" with properties like make, model, and year, and a method to display the car details.

<?php

class Car {

public $make;

public $model;

public $year;

public function \_\_construct($make, $model, $year) {

$this->make = $make;

$this->model = $model;

$this->year = $year;

}

public function displayDetails() {

echo "Car Details: <br>";

echo "Make: " . $this->make . "<br>";

echo "Model: " . $this->model . "<br>";

echo "Year: " . $this->year . "<br>";

}

}

$myCar = new Car("Toyota", "Camry", 2022);

$myCar->displayDetails();

?>

5)what is an object in OOP? Discuss how objects are instantiated from classes in PHP

An object in object-oriented programming (OOP) is a instance of a class. Each object holds its attributes or properties and supports methods or functions.

* Identity: object is separate and distinct in memory.
* State: The stored data properties or attributes.

6)Instantiate multiple objects of the "Car" class and demonstrate how to access their properties and methods.

<?php

class Car {

public $make;

public $model;

public $year;

// Constructor

public function \_\_construct($make, $model, $year) {

$this->make = $make;

$this->model = $model;

$this->year = $year;

}

//here we have Methods to display car details

public function displayDetails() {

echo "Make: $this->make, Model: $this->model, Year: $this->year<br>";

}

}

$car1 = new Car("Toyota", "Camry", 2022);

$car2 = new Car("Honda", "Civic", 2021);

$car3 = new Car("Ford", "Focus", 2020);

// Accessing properties

echo $car1->make . "<br>"; // Output: Toyota

echo $car2->model . "<br>"; // Output: Civic

// here we areCalling displayDetails() method for each object

$car1->displayDetails();

$car2->displayDetails();

$car3->displayDetails();

?>

* objects ($car1, $car2, $car3) is an independent instance of the Car class with its own set of properties.
* Properties are accessed using ->property.
* Methods are called using ->methodName().