Module 4 (Advanced php)

OOPS

- What is Object Oriented Programming?

    - OOP is an object-oriented programming technique that combines data and instructions for processing that data into an object that can be used within the program. Object-oriented programming provides concepts that help modelling complicated systems of real world into manageable software solutions.

- What are Properties of Object Oriented Systems?

    - Object
    - Class
    - Inheritance
    - Polymorphism
    - Encapsulation
    - Abstraction

- What is the Difference between class and Interface?

○ **Class**	○ **Interface**
○ A class describes the attributes and behaviour of an object.	○ An interface contains behaviour that a class implements.
○ A class may contain abstract methods, concrete methods.	○ An interface contains only abstract methods.
○ Members of a class can be public, private, protected or default.	○ All the members of the interface are public by default.

- What is Overloading?

    - Overloading is a concept used to avoid redundant code where the same method name is used multiple times but with a different set of parameters. The actual method that gets called during runtime is resolved at compile time, thus avoiding runtime errors. Overloading provides code clarity, eliminates complexity, and enhances runtime performance.

- What is the difference between Abstract classes and Interfaces?

    - Interface are similar to abstract classes. The difference between interfaces and abstract classes are:
    - Interfaces cannot have properties, while abstract classes can
    - All interface methods must be public, while abstract class methods is public or protected
    - All methods in an interface are abstract, so they cannot be implemented in code and the abstract keyword is not necessary
    - Classes can implement an interface while inheriting from another class at the same time.

- Define Constructor and Destructor?

    - Constructor:
    -
    - A constructor allows you to initialize an object's properties upon creation of the object.
    - If you create a __construct() function, PHP will automatically call this function when you create an object from a class.
    - Destructor:
    -
    - A destructor is called when the object is destructed or the script is stopped or exited.
    - If you create a __destruct() function, PHP will automatically call this function at the end of the script.

- How to load Classes in PHP?

    - PHP load classes are used for declaring its object etc. in object oriented applications. PHP parser loads it automatically, if it is registered with spl_autoload_register() function. PHP parser gets the least chance to load class/interface before emitting an error.

- How to Call Parent Constructor?

    - Parent constructors are not called implicitly if the child class defines a constructor. In order to run a parent constructor, a call to parent::__construct() within the child constructor is required. If the child does not define a constructor then it may be inherited from the parent class just like a normal class method (if it was not declared as private).

- Are Parent Constructor called implicitely when an object of class is created?

    - No, Parent constructors are not called implicitly if the child class defines a constructor.

- What Happens When Constructors is defined as Private or Protected?

    - The constructor may be made private or protected to prevent it from being called externally. If so, only a static method will be able to instantiate the class. Because they are in the same class definition they have access to private methods, even if not of the same object instance. The private constructor is optional and may or may not make sense depending on the use case.

- What is PHP Magic Methods/Functions? List them.

    - __construct()
    - __destruct()
    - __call($fun, $arg)
    - __callStatic($fun, $arg)
    - __get($property)
    - __set($property, $value)
    - __isset($content)
    - __unset($content)
    - __sleep()
    - __wakeup()
    - __toString()
    - __invoke()
    - __set_state($array)
    - __clone()
    - __debugInfo().

- Write program for Static Keyword in PHP?

    - The static keyword is also used to declare variables in a function which keep their value after the function has ended.

    -
      ```
      <?php
      function add() {
        static $n = 0;
        $n++;
        return $n;
      }

      echo add();
      ```

```
echo "<br>";
echo add();
echo "<br>";
echo add();
?>
```

- Create multiple Traits and use it in a single Class.

  -
    ```
    <?php
    trait message1 {
      public function msg1() {
        echo "This is Message 1! ";
      }
    }

    trait message2 {
      public function msg2() {
        echo "This is Message 2!";
      }
    }

    class Welcome {
      use message1, message2;
    }
    $obj = new Welcome();
    $obj->msg1();
    $obj->msg2();
    ?>
    ```

- Write PHP Script of Object Iteration?

  -
    ```
    <?php
      class person {
        public $FirstName = "Bill";
        public $MiddleName = "Terence";
        public $LastName = "Murphy";
         private $Password = "Poppy";
         public $Age = 29;
         public $HomeTown = "Edinburgh";
         public $FavouriteColour = "Purple";
      }

      $bill = new person();

      foreach($bill as $var => $value) {
         echo "$var is $value\n";
    ```

```
    }
?>
```

- Use of $this Keyword?

    - **$this** is a reserved keyword in PHP that refers to the calling object. It is usually the object to which the method belongs, but possibly another object if the method is called statically from the context of a secondary object. This keyword is only applicable to internal methods.