**Module 2 – Advanced PHP**

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

**OOP** stands for Object-Oriented Programming. // code karne ki pattern

Procedural programming is about writing procedures or functions that

perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

**Inheritance / Reusability :**

It is a concept of accessing the features of one class from another class.

The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods.

An inherited class is defined by using the [extends](https://www.w3schools.com/php/keyword_extends.asp) keyword.

**Encapsulation** in Object-Oriented Programming (OOP) in PHP refers to the bundling of data (properties) and the methods (functions) that operate on that data into a single unit, which is a class. It is a fundamental principle that promotes data hiding and controlled access to an object's internal state.

**Polymorphism** in PHP

This word is can from Greek word poly and morphism.

Poly means "many" and morphism means property which help

us to assign more than one property.

**Abstraction**

An abstract class or method is defined with the abstract keyword:

An object of an abstract class can't be made.

An abstract class is a class that contains at least one

abstract method. An abstract method is a method that is declared, but not implemented in the code.

When inheriting from an abstract class, the child class method

must be defined with the same name, and the same or a less

restricted access modifier. So,

if the abstract method is defined as protected, the child class

method must be defined as either protected or public, but not private.

**Q. Explain the structure of a class in PHP, including properties and methods**

A class is defined by using the [class](https://www.w3schools.com/php/keyword_class.asp) keyword, followed by the name of the class and a pair of curly braces ({}). All its properties and methods go inside the braces:

making group of data member(variable) and member function that called class

A class is defined by using the class keyword,

followed by the name of the class and a pair of curly braces ({}).

All its properties and methods go inside the braces:

**Q. What is an object in OOP? Discuss how objects are instantiated from classes in PHP**

Classes are nothing without objects! We can create multiple objects from a

Class. Each object has all the properties and methods defined in the class,

but they will have different property values.

Objects of a class are created using the new keyword.

**Q. Explain the concept of inheritance in OOP and how it is implemented in PHP.**

**Inheritance / Reusability :**

It is a concept of accessing the features of one class from another class.

The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods.

An inherited class is defined by using the [extends](https://www.w3schools.com/php/keyword_extends.asp) keyword.

**Q. Discuss method overloading and how it is implemented in PHP**

Overloading : Same method name with different parameter, since PHP doesn't support method overloading concept

Overloading in PHP refers to the dynamic creation of properties and methods within a class, which are handled through special "magic methods" when interacting with properties or methods that are either undeclared or not visible in the current scope.

 Unlike some other languages, PHP does not support traditional function or method overloading based on different parameter signatures directly.

**Q. Explain the concept of abstraction and the use of interfaces in PHP**

An abstract class or method is defined with the abstract keyword:

An object of an abstract class can't be made.An abstract class is a class that contains at least one abstract method. An abstract method is a method that is declared, but not implemented in the code.When inheriting from an abstract class, the child class method must be defined with the same name, and the same or a less restricted access modifier. So, if the abstract method is defined as protected, the child class method must be defined as either protected or public, but not private.

Interfaces are declared with the interface keyword:

Interfaces are characterized similarly as a class, however, only the interface keyword replaces the class phrase in the declaration and without any of the methods having their contents defined.Interfaces allow you to specify what methods a class should implement.

PHP - Interfaces vs. Abstract Classes

Interfaces are similar to abstract classes. The difference between interfaces and abstract classes are:Interfaces cannot have properties, while abstract classes canAll interface methods must be public, while abstract class methods is public or protectedAll methods in an interface are abstract, so they cannot be implemented in code and the abstract keyword is not necessary.

**Q. What is a constructor in PHP? Discuss its purpose and how it is used**

A constructor allows you to initialize an object's properties upon creation of the object. Also called magic function If you create a \_\_construct() function, PHP will automatically call this function when you create an object from a class.Notice that the construct function starts with two underscores (\_\_)!

**Q. Explain the role of a destructor in PHP and when it is called.**

In PHP, destructor method is named as \_\_destruct.

During shutdown sequence too, objects will be destroyed.

Destructor method doesn't take any arguments, neither does it return any data type

**Q. Define magic methods in PHP. Discuss commonly used magic methods like \_\_get(), \_\_set(), and \_\_construct().**

Advance Function

Magic Methods / Build in Function/predefined

Magic methods are special methods which override PHP's default's

action when certain actions are performed on an object.

The \_\_get and \_\_set magic methods in PHP are used to intercept calls to get or set the value of inaccessible properties of an object. They allow you to define custom behaviors when getting or setting the value of a property that is not directly accessible.

A constructor allows you to initialize an object's properties

upon creation of the object. Also called magic function

If you create a \_\_construct() function, PHP will automatically call

this function when you create an object from a class.

**Q. Explain the scope resolution operator (::) and its use in PHP**

In PHP, the Scope Resolution Operator, represented by the double colon ::, is used to access static members (properties and methods) and constants within a class, or to call methods of a parent class from a child class

used to access members of a class, namespace, or global scope, resolving ambiguities when a local variable or function has the same name as an identifier in a wider scope.

It is essential for calling global functions, accessing static or class members, defining class methods outside the class body, specifying which namespace to use, and accessing members of nested scopes or inheritance.

**Q. Define traits in PHP and their purpose in code reuse.**

a child class can inherit only from one single parent.

So, what if a class needs to inherit multiple behaviors?

OOP traits solve this problem.

Traits are declared with the trait keyword: as class

To use a trait in a class, use the use keyword: // for inheritance

Traits are used to declare methods that can be used in

multiple classes.

Traits can have methods and abstract methods that can be used in

multiple classes, and the methods can have any access modifier

(public, private, or protected).

**Q. Discuss the visibility of properties and methods in PHP (public, private, protected)**

Properties and methods can have access modifiers which control where they can be accessed.

There are three access modifiers:

* [public](https://www.w3schools.com/php/keyword_public.asp) - the property or method can be accessed from everywhere. This is default
* [protected](https://www.w3schools.com/php/keyword_protected.asp) - the property or method can be accessed within the class and by classes derived from that class
* [private](https://www.w3schools.com/php/keyword_private.asp) - the property or method can ONLY be accessed within the class

In the following example we have added three different access modifiers to three properties (name, color, and weight). Here, if you try to set the name property it will work fine (because the name property is public, and can be accessed from everywhere). However, if you try to set the color or weight property it will result in a fatal error (because the color and weight property are protected and private):

**Q. Explain type hinting in PHP and its benefits**

Type hinting in PHP involves explicitly declaring the expected data type for function parameters, return values, and class properties. This allows PHP to enforce these types at runtime, leading to more robust and predictable code.

How it works:

You specify the desired type before the variable name in function parameters, after the closing parenthesis for return types, and before the property name in class definitions.

**Benefits of Type Hinting:**

* **Improved Code Readability and Clarity:**

Type hints clearly communicate the expected data types, making it easier for developers to understand the purpose and usage of functions and classes.

* **Reduced Errors and Enhanced Code Integrity:**

By enforcing type checks at runtime, type hinting helps catch type-related errors early, preventing unexpected behavior and improving the reliability of your code.

* **Better IDE Support and Autocompletion:**

Modern IDEs leverage type hints to provide more accurate code suggestions, autocompletion, and static analysis, boosting developer productivity.

* **Easier Debugging:**

When a type mismatch occurs, PHP throws a clear TypeError, pinpointing the exact location and nature of the issue, which simplifies debugging.

* **Facilitates Refactoring:**

With explicit type declarations, refactoring code becomes safer as changes to data types are immediately identified, reducing the risk of introducing new bugs.

* **Enforces Design Choices:**

Type hinting encourages more thoughtful design by requiring developers to be explicit about the data structures and types used throughout their application.

**Q. Discuss the purpose of the final keyword in PHP and how it affects classes and methods**

PHP introduces the final keyword, which prevents child classes from

overriding a method by prefixing the definition with final.

If the class itself is

being defined final then it cannot be extended.

**Purpose:**

The primary purpose of the final keyword is to prevent modification of specific classes or methods within an inheritance hierarchy. This is crucial in scenarios where:

* **Security or Design Integrity:**

Certain core functionalities or sensitive operations should not be altered by child classes. Declaring them final ensures their intended behavior.

* **API Stability:**

When creating libraries or frameworks, making certain classes or methods final guarantees that external users cannot extend or override them, thus maintaining a consistent and predictable API.

* **Preventing Unintended Consequences:**

Uncontrolled inheritance and overriding can lead to unexpected behavior or bugs in complex systems. final helps to prevent such issues by limiting modification points.

**Effects on Classes:**

When a class is declared final, it cannot be extended by any other class. Attempting to do so will result in a fatal error.

This ensures that the final class's structure and behavior remain exactly as defined, preventing any further specialization or modification through inheritance.

**Effects on Methods:**

When a method within a class is declared final, it cannot be overridden by any child class that extends the parent class. Attempting to override a final method will also result in a fatal err

This ensures that the final method's implementation remains consistent across the entire inheritance chain, regardless of how child classes might extend other parts of the parent class.

**Q. Explain the importance of email security and common practices to ensure secure email transmission.**

Email security is vital to protect sensitive data from cyber threats like phishing, malware, and unauthorized access, ensuring confidentiality, integrity, and compliance. Common security practices include using strong, unique passwords, enabling multi-factor authentication (MFA), implementing [email encryption](https://www.google.com/search?rlz=1C1FKPE_enIN995IN995&cs=0&sca_esv=70a532a702b7bdb6&sxsrf=AE3TifNLE6kM3sVNiHM0BYUUkt71jmU50g%3A1758883233116&q=email+encryption&sa=X&ved=2ahUKEwihpY_SnvaPAxWcT2wGHfJvF60QxccNegQIBRAB&mstk=AUtExfBFx8puJAiZH79ed5heTuj6I-q7MQ8TMxqaVK3ftkHe2Z4pLEye61aDawpRrG4DCAUjsjChNFhddW03y00mLQuqeht7tOoq6aEf60b6F0KRWlC3llfPFQEdmcHcaWA_utCy1jY9xKIh9ehfKXxNUSArEfczYQse9iC62pHFFj2zY7Q&csui=3), being cautious of suspicious links and attachments, keeping software updated, and educating users to recognize and avoid threats.

**Why Email Security is Important**

* **Protects Sensitive Information:**

Emails often carry confidential data, such as financial details, personal information, and intellectual property, making email security essential for preventing breaches.

* **Combats Cyber Threats:**

Email is a primary vector for attacks like phishing, malware, and ransomware, which can compromise systems, steal data, and cause significant financial loss.

* **Maintains Confidentiality and Integrity:**

Security measures ensure that only intended recipients can access email content and that the information remains unchanged during transmission.

* **Ensures Business Compliance:**

Many industries are subject to regulations that require the protection of sensitive data, making robust email security a necessity for compliance.

* **Prevents Identity Theft and Fraud:**

By securing email accounts and communications, businesses and individuals can prevent attackers from stealing credentials or conducting fraudulent activities.

**Common Practices for Secure Email Transmission**

* **Use Strong, Unique Passwords:**

Create complex passwords for your email accounts and change them regularly to prevent unauthorized access.

* **Enable**[**Multi-Factor Authentication**](https://www.google.com/search?rlz=1C1FKPE_enIN995IN995&cs=0&sca_esv=70a532a702b7bdb6&sxsrf=AE3TifNLE6kM3sVNiHM0BYUUkt71jmU50g%3A1758883233116&q=Multi-Factor+Authentication&sa=X&ved=2ahUKEwihpY_SnvaPAxWcT2wGHfJvF60QxccNegQILxAB&mstk=AUtExfBFx8puJAiZH79ed5heTuj6I-q7MQ8TMxqaVK3ftkHe2Z4pLEye61aDawpRrG4DCAUjsjChNFhddW03y00mLQuqeht7tOoq6aEf60b6F0KRWlC3llfPFQEdmcHcaWA_utCy1jY9xKIh9ehfKXxNUSArEfczYQse9iC62pHFFj2zY7Q&csui=3)**(MFA):**

Add an extra layer of security by requiring a second verification step, such as a code from your phone, to log in.

* **Utilize Email Encryption:**

Use encryption services to scramble the content of your emails, ensuring only the intended recipient can read them.

* **Be Wary of Suspicious Emails:**

Avoid clicking on dubious links or opening attachments in emails from unknown senders or that seem unusual.

* **Keep Software Updated:**

Regularly update your operating system, email client, and antivirus software to patch vulnerabilities that attackers could exploit.

* **Educate Users:**

Train individuals to recognize phishing attempts and other social engineering tactics, as human awareness is a critical part of security.

**Q. Discuss file handling in PHP, including opening, reading, writing, and closing files**

A better method to open files is with the fopen() function. This function gives you more options than the readfile() function.

The first parameter of fopen() contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened. The following example also generates a message if the fopen() function is unable to open the specified file:

The fread() function reads from an open file.

The first parameter of fread() contains the name of the file to read from and the second parameter specifies the maximum number of bytes to read.

The following PHP code reads the "webdictionary.txt" file to the end:

The fclose() function is used to close an open file.

**Q. Explain how to send emails in PHP using the mail() function and the importance of validating email addresses**

The mail() function in PHP provides a basic way to send emails. It requires the recipient's email address, subject, and message body. Optional parameters include additional headers and command-line parameters for the sendmail program.

**Importance of Validating Email Addresses**

Validating email addresses before sending emails is crucial for several reasons:

* **Preventing Bounces and Improving Deliverability:**

Sending emails to invalid addresses leads to bounces, which can negatively impact your sender reputation and cause your legitimate emails to be flagged as spam. Validation ensures emails are sent to existing addresses, improving deliverability.

* **Maintaining Data Quality:**

Validating email addresses helps maintain clean and accurate user data in your database, preventing the accumulation of invalid or outdated information.

* **Reducing Server Load and Costs:**

Sending emails to invalid addresses consumes server resources and can incur costs if you are using a third-party email service. Validation reduces this unnecessary overhead.

* **Enhancing User Experience:**

Users appreciate receiving relevant communication. Sending emails to invalid addresses can lead to confusion or frustration for users who may have mistyped their address.

* **Security and Spam Prevention:**

Validation can help prevent malicious actors from using your system to send spam or engage in other harmful activities by ensuring only legitimate email addresses are processed.

**Q. Discuss the Model-View-Controller (MVC) architecture and its advantages in web development.**

MVC is abbreviated as Model View Controller is a design pattern

created for developing applications specifically web applications.

As the name suggests, it has three major parts. The traditional software

design pattern works in an "Input - Process - Output" pattern whereas

MVC works as "Controller -Model - View" approach. With the emergence

of the MVC model, creation of application takes different aspects

individually into consideration. These aspects of the application are:

Let us discuss the three components of MVC in brief:

**Model:** The Model encloses the clean application related data. But the model does not deal with any logic about how to present the data.

MODEL CONNECTED WITH Database + Logic (crud ins/ipd/del/sel)

**Controller:** The Controller is in between the model and the view element. It listens to all the incident and actions triggered in the view and performs an appropriate response back to the events.

**View:** The View element is used for presenting the data of the model to the user.This element deals with how to link up with the model's data but doesn't provide any logic regarding what this data all about or how users can use these data.

Advantages of MVC in Web Development:

* **Separation of Concerns:**

MVC promotes a clear separation between the data (Model), presentation (View), and logic (Controller). This modularity makes the codebase more organized, easier to understand, and simpler to maintain.

* **Enhanced Maintainability:**

With distinct components, changes to one part of the application are less likely to affect others, simplifying debugging and updates.

* **Improved Testability:**

Each component can be tested independently, facilitating unit testing and enabling Test-Driven Development (TDD).

* **Parallel Development:**

Different developers can work simultaneously on the Model, View, and Controller components without significant conflicts, accelerating the development process.

* **Code Reusability:**

Models and Controllers can often be reused across different views or even in other applications, reducing development time and effort.

* **Flexibility for Multiple Views:**

The separation allows for the creation of multiple views for a single model, catering to different user interfaces or device types.

* **SEO-Friendly URLs:**

MVC frameworks often provide robust routing mechanisms, enabling the creation of clean, descriptive, and SEO-friendly URLs.

* **Scalability:**

The modular nature of MVC makes it easier to scale applications by adding or modifying individual components as needed.

**Q. Explain how to connect PHP to a MySQL database using mysqli or PDO.**

Connecting PHP to a MySQL database can be achieved using either the mysqli extension or PHP Data Objects (PDO). Both offer secure and efficient ways to interact with your database.

1. Using mysqli (Object-Oriented Style):

The mysqli extension provides an object-oriented interface for interacting with MySQL.

**Code:**

<?php

$servername = "localhost"; // Or your database host

$username = "your\_username";

$password = "your\_password";

$dbname = "your\_database\_name";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully using mysqli!";

// Perform database operations here (e.g., queries, updates)

// Close connection

$conn->close();

?>

2. Using PDO (PHP Data Objects):

**Code:**

<?php

$servername = "localhost"; // Or your database host

$username = "your\_username";

$password = "your\_password";

$dbname = "your\_database\_name";

try {

$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);

// Set the PDO error mode to exception

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

echo "Connected successfully using PDO!";

} catch(PDOException $e) {

echo "Connection failed: " . $e->getMessage();

}

// Perform database operations here (e.g., queries, updates)

// Close connection (PDO connections are automatically closed when the script ends,

// but you can explicitly set $conn = null; if needed)

$conn = null;

?>

PDO offers a consistent interface for accessing various databases, not just MySQL, promoting more portable code.

Key Considerations:

* **Security:**

Always use prepared statements with placeholders when executing queries to prevent SQL injection vulnerabilities, regardless of whether you use mysqli or PDO.

* **Error Handling:**

Implement robust error handling to gracefully manage connection failures and other database-related issues.

* **Credentials:**

Store your database credentials securely and avoid hardcoding them directly in publicly accessible files in a production environment.

**Q. Define SQL injection and its implications on security.**

SQL injection is a code injection technique that might destroy your database.

SQL injection is one of the most common web hacking techniques.

SQL injection is the placement of malicious code in SQL statements,

via web page input.

**SQL in Web Pages**

SQL injection usually occurs when you ask a user for input,

like their username/userid, and instead of a name/id,

the user gives you an SQL statement that you will

unknowingly run on your database.

Many web applications are connected to a database.

The database holds all the information the web application

wish to store and use.

SQL Injection is a technique which allows attackers to

manipulate the SQL

("Structured Query Language")

the developer of the web application is using.

This typically happens because of lack of data sanitization.

SQL is used regularly by developers to access database resources.

$first\_name=real\_escape\_string($\_REQUEST['firstname']);

This both function convert query into string than we avoid sql injection

**Q. Explain the differences between sessions and cookies in PHP.**

Sessions and cookies in PHP both serve to maintain state in the stateless HTTP protocol, but they differ fundamentally in where and how data is stored.

Cookies:

Storage Location: Stored on the client-side, within the user's web browser, typically as small text files.

**Data Content**: Store small pieces of information, such as user preferences, login status indicators, or tracking data.

**Lifetime**: Can be set to expire after a specific duration (persistent cookies) or when the browser is closed (session cookies).

**Security**: Less secure as they are stored on the client and can be accessed or potentially modified by the user or client-side scripts. Sensitive data should not be stored directly in cookies without encryption.

**Size Limit**: Have a limited size, typically around 4KB per cookie.

**Sessions:**

**Storage Location**: Stored on the server-side, typically in temporary files or a database.

**Data Content**: Store more extensive and potentially sensitive user data, such as login information, shopping cart contents, or user-specific variables.

**Lifetime**: Typically expire when the user closes their browser or after a server-defined inactivity timeout.

**Security**: Generally more secure as the actual data resides on the server. Only a unique session ID is exchanged with the client, usually via a session cookie.

**Size Limit**: Can hold a larger amount of data, limited by server resources rather than a strict per-item size limit.

**Q. Discuss file upload functionality in PHP and its security implications.**

File upload in PHP allows you to upload files with different extensions to the server. We can use HTML forms and enable the users to upload files to the server. These files are stored in a temporary directory unless moved to a target location for permanent storage

Corporate web applications commonly need to accept file uploads such as images, word documents and other file types from their users. However, it’s possible that these files may contain malicious content or malware. File security protects corporate servers against these malicious uploads. By inspecting uploaded files for malicious content, it can block malware from being uploaded to corporate servers via web applications.

File security helps to limit the security risks posed by file upload functionality on corporate web apps. Some file security best practices include the following:

**Inspect Uploaded Files**: Files uploaded to corporate web applications can have a wide range of potentially malicious content. Inspecting this traffic before it reaches the target web app is essential to threat prevention.

**Subscribe to Threat Intelligence Feeds**: Effective file security is dependent on the ability to accurately identify malicious uploaded files. Access to high-quality threat intelligence is vital to detecting malware associated with the latest threat campaigns.

**Tune Security Thresholds**: A file security risk score represents the probability that an uploaded file is malicious and is compared to a threshold to decide whether or not

the file should be blocked. This threshold should be tuned to ensure corporate security without negatively impacting the user experience.