Unit I: Introduction to PHP

PHP for Web Development

- **Definition**: PHP (Hypertext Preprocessor) is a server-side scripting language designed for web development.
- **Key Features**:
    - Open-source and free to use.
    - Compatible with various platforms like Windows, Linux, macOS.
    - Efficient for creating dynamic and interactive web pages.
    - Can be embedded into HTML, making it easy to use.
    - Supports a wide range of databases, including MySQL.
    - Extensive community support and vast libraries.
    - Provides high flexibility and scalability for web development.
    - Offers robust security features with built-in functionalities like input validation and encryption.

History and Future Scope of PHP

- **History**:
    - Developed in 1994 by Rasmus Lerdorf.
    - Initially created as a set of Common Gateway Interface (CGI) scripts for personal use.
    - Officially released as PHP/FI (Forms Interpreter) in 1995.
    - Evolved through multiple versions, with PHP 8 being the latest major release (as of 2023).
    - Gained popularity with the introduction of dynamic websites and database integration.
- **Future Scope**:
    - Continues to be widely used for content management systems (CMS) like WordPress, Joomla, and Drupal.
    - Increasing integration with modern frameworks like Laravel and Symfony.
    - Enhanced performance and security features in newer versions ensure its relevance in web development.
    - Ongoing advancements in PHP libraries and tools enable faster development processes.
    - The language remains a choice for building RESTful APIs and backend systems.

Relationship between PHP, MySQL, and Apache

- **PHP**: A scripting language used to create server-side logic.
- **MySQL**: A relational database management system (RDBMS) used to store and retrieve data for web applications.
- **Apache**: A web server software that serves web pages to users.
- **Integration**:
  - Apache handles requests from users and passes them to PHP scripts.
  - PHP processes the logic and interacts with MySQL to fetch or update data.
  - The combined output is sent back to the Apache server, which delivers it to the user's browser.
  - This combination, often referred to as the LAMP stack (Linux, Apache, MySQL, PHP), is popular for building robust web applications.

Structure and Syntax of PHP

PHP Tags:

```
<?php
  // PHP code goes here
?>
```

Case Sensitivity: Keywords are not case-sensitive, but variables are case-sensitive.

Basic Syntax:

```
echo "Hello, World!";
```

Comments:

Single-line: // or #

Multi-line: /* */

Semicolon: Every statement in PHP ends with a semicolon (;).

Whitespace: PHP ignores extra spaces or tabs, making the code flexible in formatting.

PHP Variables and Its Data Types

- **Variables**:
  - Begin with a $ symbol (e.g., $variableName).
  - Must start with a letter or underscore.
  - Dynamically typed; no need to declare the type.

  - Variables are used to store data, which can be accessed and modified during script execution.
  - Example:
    - $name = "John";
    - $age = 25;

- **Data Types**:
  - Scalar: Integer, Float, String, Boolean
  - Compound: Array, Object
  - Special: NULL, Resource
  - Examples:
    - $intVar = 10; // Integer
    - $floatVar = 10.5; // Float
    - $stringVar = "Hello"; // String
    - $boolVar = true; // Boolean

Type Casting and Garbage Value

- **Type Casting**:
  - Explicit conversion of data types using syntax like (int), (float), (string).
  - Example:
    - $x = "5"; // string
    - $y = (int)$x; // integer
- **Garbage Value**:
  - PHP manages memory automatically; garbage values (unreferenced memory) are cleaned by the Garbage Collector.
  - Variables no longer in use are automatically freed to optimize memory usage.

Control Statements

- Used to control the flow of execution based on conditions.

If Statement:

```
if (condition) {
        // code to execute
}
```

If...Else Statement:

```
    if (condition) {
        // code if condition is true
    } else {
        // code if condition is false
    }
```

Nested If Statement:

```
    if (condition1) {
        if (condition2) {
        // code if both conditions are true
        }
    }
```

Switch Statement:

```
    switch (variable) {
        case value1:
        // code for value1
        break;
        case value2:
        // code for value2
        break;
        default:
        // default code
    }
```

- **Best Practices for Control Statements**:
    - Use meaningful and clear condition expressions.
    - Keep the code block within conditions concise for readability.
    - Avoid deeply nested conditions to simplify the code structure.

Looping Statements

- Used to execute a block of code repeatedly based on a condition.

For Loop:

```
for (initialization; condition; increment/decrement) {
        // code to execute
}
```

- Example:

```
  for ($i = 1; $i <= 5; $i++) {
        echo $i . " ";
  }
```

While Loop:

```
while (condition) {
        // code to execute
}
```

- Example:

```
$i = 1;
while ($i <= 5) {
        echo $i . " ";
        $i++;
}
```

Do...While Loop:

```
do {
        // code to execute
} while (condition);
```

- Example:

```
$i = 1;
do {
        echo $i . " ";
        $i++;
        } while ($i <= 5);
```

For Each Loop:

- **Used specifically for iterating over arrays.**

```
foreach ($array as $value) {

        // code to execute

}
```

- Example:

```
$colors =array["Red", "Green", "Blue"];
foreach ($colors as $color) {
        echo $color . " ";
}
```

- **Tips for Looping Statements**:
  - Ensure conditions are well-defined to prevent infinite loops.
  - Use break to exit a loop prematurely when necessary.
  - Use continue to skip the rest of the loop's body and proceed to the next iteration.
  - For improved readability, avoid deeply nested loops.

PHP Questions and Answers

5 Marks Questions

1. **Explain the key features of PHP that make it suitable for web development. Answer**:
   - Open-source and free to use.
   - Compatible with multiple platforms (Windows, Linux, macOS).
   - Allows embedding into HTML for dynamic content creation.
   - Supports a wide range of databases, including MySQL.
   - Has extensive libraries and community support.
2. **What is the relationship between PHP, MySQL, and Apache in web development? Answer**:
   - PHP is the server-side scripting language used for backend logic.
   - MySQL is the database system used for storing and managing data.
   - Apache is the web server software that processes client requests and delivers responses.
   - Together, they form the LAMP stack for developing web applications.

3. **Describe the structure and syntax of PHP.**
   Answer:
   - PHP code is enclosed within <?php ... ?> tags.
   - Statements end with a semicolon (;).
   - Comments can be single-line (//, #) or multi-line (/* */).
   - Variables start with $ and are case-sensitive.
   - Example:

     ```
     <?php
     echo "Hello, World!";
     ?>
     ```
4. **Explain the concept of type casting in PHP with an example.**
   Answer:
   - Type casting involves converting a variable from one data type to another.
   - Example:

     ```
     $x = "10"; // string
     $y = (int)$x; // integer
     echo $y; // Outputs: 10
     ```
5. **Write and explain the syntax for the 'switch' statement in PHP.**
   Answer:
   - The switch statement is used to execute different blocks of code based on a variable's value.
   - **Syntax:**

     ```
     switch ($variable) {
         case value1:
         // Code for value1
         break;
         case value2:
         // Code for value2
         break;
         default:
         // Default code
     }
     ```

     Example:

     ```
     $day = "Monday";
     switch ($day) {
             case "Monday":
             echo "Start of the work week!";
     ```

```
                break;
                default:
                echo "Not Monday.";
        }
```

Easy Questions

1. **What does PHP stand for?**
   Answer: PHP stands for Hypertext Preprocessor.
2. **What symbol is used to start variables in PHP?**
   Answer: The $ symbol is used to start variables in PHP.
3. **Is PHP case-sensitive when it comes to variables?**
   Answer: Yes, PHP variables are case-sensitive.
4. **Write an example of a single-line comment in PHP.**
   Answer: 5. // This is a single-line comment
6. **What is the purpose of the echo statement in PHP?**
   Answer: The echo statement is used to output data to the browser.

Fill in the Blanks

1. PHP stands for **Hypertext Preprocessor**.
2. PHP scripts are executed on the **server**.
3. Variables in PHP start with the **$** symbol.
4. The for loop in PHP requires an **initialization, condition**, and **increment/decrement**.
5. PHP is widely used for building **dynamic** and **interactive** websites.
6. PHP was created by **Rasmus Lerdorf** in 1994.
7. PHP supports a database integration system called **MySQL**.
8. A PHP script starts with <?php and ends with ?>.
9. The foreach loop is specifically used for iterating over **arrays**.
10. The PHP function to print output is **echo**.

True/False

1. PHP is a client-side scripting language. **False**
2. PHP variables are case-sensitive. **True**
3. PHP can be embedded directly into HTML. **True**
4. The switch statement can handle multiple cases in PHP. **True**
5. PHP does not support databases other than MySQL. **False**
6. PHP code can only be executed on Linux servers. **False**
7. PHP is an open-source programming language. **True**
8. PHP scripts need to be compiled before running. **False**
9. The while loop executes at least once, even if the condition is false. **False**
10. The do...while loop executes at least once, even if the condition is false. **True**