

PHP Module 4 (Previous Year QP Discussion)

Sem 4 BCA/B.Sc. Computer Science MGU



For Notes download BCA Resources App

Module 4 Overview

- PHP functions, PHP form, Passing information between pages, \$_GET, \$_POST, \$_REQUEST. String functions, include and require, session and cookie management, error handling in PHP, Object Oriented Programming using PHP



1. What is PHP Session? [2019] [2 Marks]

- PHP Session is a mechanism used in web development to maintain stateful information across multiple requests from the same user. It allows data to be stored on the server and associated with a unique session identifier, usually stored in a cookie on the user's browser, to provide personalized content and maintain user-specific information throughout their session.
- you can store data to `$_SESSION` (Global Variable), an associative array



2. Differentiate function overloading & overriding? [2019] [2 Marks]

- Function Overloading:
 - Function overloading is a concept in object-oriented programming where multiple functions with the same name exist within a class, but they differ in the number or types of parameters they accept.
 - The selection of the appropriate function to execute is determined by the number or types of arguments passed to the function during its invocation.
- Function Overriding:
 - Function overriding is a concept in object-oriented programming where a subclass provides a specific implementation for a method that is already defined in its superclass.
 - The method in the subclass "overrides" the implementation of the method in the superclass, providing a specialized behavior for that method in the context of the subclass. This allows for polymorphism, where the same method call can have different behaviors depending on the object's actual type at runtime.



3. Explain any 5 string handling functions used in PHP? [2019] [5 Marks]

1. `strlen()` : This function is used to get the length of a string. It returns the number of characters in the given string, including spaces and special characters.

```
$string = "Hello, World!";  
$length = strlen($string); // $length will be 13
```

2. `strrev()` : reverses a string.

```
<?php  
echo strrev("Hello world!"); // outputs !dlrow olleH  
?>
```

3. `str_replace()` : used to find and replace occurrences of a substring within a string.

```
$string = "The quick brown fox jumps over the lazy dog."  
$newString = str_replace("fox", "cat", $string); // $newString will be "The quick brown cat jumps over the lazy dog."
```

4. `strtoupper()` : converts all characters in a string to uppercase.

```
$string = "Hello, World!";  
$uppercaseString = strtoupper($string); // $uppercaseString will be "HELLO, WORLD!"
```

5. `strtolower()` : converts all characters in a string to lowercase.

```
$string = "Hello, World!";
```

6. `$lowercaseString = strtolower($string);` // \$lowercaseString will be "hello, world!"

4. Explain Exception handling? [2019] [5 Marks]

- Exception handling in PHP is a mechanism that allows developers to manage and respond to runtime errors or exceptional situations gracefully. It helps in controlling the flow of a program when unexpected issues occur, providing a more robust and maintainable codebase. Exception handling involves the use of three main components: try, catch, and throw.
1. try block: The try block is used to wrap the code that might raise an exception. It allows you to specify a section of code that should be monitored for exceptions.
 2. catch block: The catch block follows the try block and is used to handle any exceptions that are thrown within the corresponding try block. You can have multiple catch blocks to handle different types of exceptions.
 3. throw statement: The throw statement is used to explicitly throw an exception. When an exceptional situation occurs in the try block, the code can use throw to generate a new exception and pass it to the appropriate catch block for handling.



4. Explain Exception handling? [2019] [5 Marks]

```
try {  
    $numerator = 10;  
    $denominator = 0;  
  
    if ($denominator === 0) {  
        throw new Exception("Division by zero is not allowed.");  
    }  
    $result = $numerator / $denominator;  
    echo "Result: " . $result;  
} catch (Exception $e) {  
    echo "Exception caught: " . $e->getMessage();  
}
```



5. What is the use of substr() function in PHP? [2020] [2 Marks]

The substr() function in PHP is used to extract a substring from a given string. It allows you to obtain a portion of a string based on a specified starting index and, optionally, a length.

```
$string = "Hello, World!";
```

```
$substring = substr($string, 0, 5); // $substring will be "Hello"
```



6. What are cookies? [2020] [2 Marks]

Cookies in PHP are small pieces of data stored on the user's browser that allow web applications to remember information between different page visits. They are commonly used to maintain stateful information about the user or to store user preferences.

- **Data Storage:** Cookies can store small amounts of data, typically up to 4KB in size. They are sent to and stored on the user's browser when the web server sets them using the `setcookie()` function in PHP.
- **Persistence:** Cookies can have an expiration time set by the server. If an expiration time is not provided, the cookie will remain on the user's browser until it is manually deleted or the browser session ends.



7. Differentiate isset() & unset() functions in PHP? [2020] [5 Marks]

isset() function: used to check if a variable is set and not null.

Usage: It takes one or more variable names as arguments and returns a boolean value (true or false) indicating whether the variable is defined and has a non-null value.

Return Value: It returns true if the variable is defined and not null; otherwise, it returns false.

```
$name = "John";  
$age = 30;  
if (isset($name)) {  
    echo "Name is set: " . $name; // Output: Name is set: John  
}  
  
if (isset($email)) {  
    // This block won't be executed as $email is not defined.  
}
```



7. Differentiate isset() & unset() functions in PHP? [2020] [5 Marks]

unset() function: used to destroy or unset a variable or an element within an array.

Usage: It takes one or more variable names as arguments and removes the specified variables, freeing the associated memory.

Return Value: Unlike isset(), unset() does not return any value

```
$name = "John";  
$age = 30;  
unset($age);  
if (isset($name)) {  
    echo "Name is set: " . $name; // Output: Name is set: John  
}  
if (isset($age)) {  
    // This block won't be executed as $age has been unset.  
}
```



8. Which are the error level used in PHP? [2020] [5 Marks]

1. **E_ERROR:** This is a fatal error that halts the script's execution. It indicates a critical issue that usually prevents the script from continuing. Examples include calling an undefined function or division by zero.
2. **E_WARNING:** Warnings are non-fatal errors that don't stop script execution but indicate potential problems in the code. For example, using an uninitialized variable or including a file that doesn't exist.
3. **E_NOTICE:** Notices are non-fatal messages that indicate potential issues, but the script can continue without any interruption. It is used to warn about situations like accessing undefined array keys or variables.
4. **E_PARSE:** Parse errors occur during the compilation of PHP code. They indicate syntax errors that prevent the script from being parsed and executed correctly.
5. **E_DEPRECATED:** Deprecated warnings inform developers that a particular feature or function is no longer recommended and might be removed in future PHP versions. It encourages the use of alternative approaches.



9. How inheritance is implemented in PHP? Explain how base class methods & properties are accessed? Give Example [2020] [15 Marks]

- Inheritance is a fundamental object-oriented programming concept that allows a class (the subclass or derived class) to inherit properties and methods from another class (the base class or parent class).
- In PHP, inheritance is implemented using the extends keyword. When a class extends another class, it automatically inherits all the public and protected methods and properties from the parent class.
- PHP don't support multiple inheritance.
- private properties and methods cannot be accessed directly from subclasses at all, and they are only accessible within the class they are defined in.



10. What is the use of explode() function. [2021] [2 Marks]

Breaks a string into array.

```
<?php  
$str = "Anto Josu";  
print_r (explode(" ", $str));  
?>
```

Output: Array ([0] => Anto [1] => Josu)



11. List out the parameters of setcookie() function. [2021] [2 Marks]

- name (string): The name of the cookie. It is the identifier that will be used to access the cookie value later.
- value (string): The value to be stored in the cookie. This can be any string or data that you want to persistently store.
- secure (bool): Optional. If set to true, the cookie will only be sent over secure HTTPS connections. If set to false, it will be sent over both HTTP and HTTPS connections.
- httponly (bool): Optional. If set to true, the cookie will be accessible only through the HTTP protocol and cannot be accessed through client-side scripts such as JavaScript. This provides a level of protection against cross-site scripting (XSS) attacks

// Set a cookie with name 'user_id', value '12345', and expiration time of one week (7 days)

```
setcookie('user_id', '12345', time() + (7 * 24 * 60 * 60), '/', 'example.com', true, true);
```



12. Write a short note on visibility in PHP class [2021] [5 Marks]

- In PHP, visibility properties are used to control the access levels of class properties and methods from different parts of the code. There are three visibility modifiers that you can use in a PHP class:
- Public (public): A public property or method can be accessed from anywhere, both inside and outside the class. It has no access restrictions.
- Protected (protected): A protected property or method can only be accessed from within the class itself or from its subclasses (child classes). It cannot be accessed from outside the class hierarchy.
- Private (private): A private property or method can only be accessed from within the class itself. It is not accessible from subclasses or from outside the class.



13. Explain different categories of errors in PHP. What are the different error handling mechanism? [2021] [15 Marks]

- Different categories of errors : refer Error level question which we discussed before
- Error handling mechanism: check the PHP Exception question's answer & mention about die() function

