PHP Bank questions second Term

1. Remembering (Recall facts and basic concepts)

1. What did the acronym PHP originally stand for when it was created?
2. Who is the Danish-Canadian programmer that created the first iteration of PHP in 1994?
3. What is the current recursive definition for the acronym PHP?
4. Which two developers rewrote the PHP parser in 1998 to launch PHP 3.0?
5. What is the name of the core scripting engine for PHP developed in 2000?
6. Which version of PHP introduced full support for Object-Oriented Programming (OOP)?
7. What is the standard file extension used for PHP files?
8. Which PHP data type represents a non-decimal number between -2,147,483,648 and 2,147,483,647?
9. What is an array in PHP?
10. Name the three specific types of arrays supported in PHP.
11. Which operator is used for string concatenation in PHP?
12. What are the opening and closing syntax tags for a PHP script?
13. Name three jumping control statements used in PHP.
14. Which built-in PHP function is utilized to permanently delete a file?
15. What superglobal array does PHP use to handle file uploads?
16. What does the "A" in the XAMPP acronym stand for?
17. Which keyword is required to create a new object instance from a class in PHP?

2. Understanding (Explain ideas or concepts)

18. Explain the primary difference between how a static web server and a dynamic web server deliver content.
19. How does the == comparison operator differ from the === identical operator?
20. Describe the purpose of a constructor function in Object-Oriented Programming.
21. What distinguishes an indexed array from an associative array?
22. In the context of loops, explain the functional difference between the `break` and `continue` statements.
23. What is the core purpose of implementing a `try-catch` block in PHP?

24. Explain the difference between an error and an exception during program execution.
25. How do session variables differ from cookies regarding where their data is stored?
26. What does the `fopen()` function do, and how does it behave specifically in "w" mode?
27. Describe the primary role of the Zend Engine in the PHP ecosystem.
28. Why is it beneficial to leave comments in your PHP scripts?
29. What happens to the program's execution flow when the `exit` statement is triggered?
30. Explain what a logical error is and why the system does not produce a warning message for it.
31. How does file downloading function conceptually between a client's browser and a web server?
32. What is encapsulation in the context of PHP's Object-Oriented Programming?
33. Differentiate the accessibility of `public`, `private`, and `protected` class properties.
34. Why might a web developer use the `md5()` function when building a site visitor counter?

3. Applying (Use information in new situations)

35. Write a PHP function with arguments that calculates the total cost of a Smart Ride Rwanda trip.
36. Use a `foreach` loop to display student names from an associative array for an attendance system.
37. Write a `while` loop that counts upwards from 1 to 10.
38. Apply the `str_replace()` function to change the word "world" to "Dolly" in the string "Hello world!".
39. Write a `switch` statement that outputs the correct day of the week based on an integer input between 1 and 7.
40. Create a multi-dimensional array to catalog three books for the New Generation Academy library management system.
41. Apply the `setcookie()` function to securely remember a user's login preference for one hour.
42. Write a PHP snippet that opens a file named "pulse_logs.txt" in append mode ("a") and adds a new line of text.

43. Use the `move_uploaded_file()` function to successfully store a temporary file into a specific "uploads/" directory.
44. Write an `if...elseif...else` statement that checks a student's marks and assigns a passing or failing grade.
45. Calculate the total pages needed to paginate 100 database items if you display 10 per page, utilizing the `ceil()` function.
46. Apply `mysqli_connect()` to establish a connection to a local database named "my_database".
47. Write a `do-while` loop that successfully calculates the sum of numbers from 1 to 10.
48. Apply the `strlen()` function to calculate the length of the string "Digital Solutions".
49. Instantiate a new object from a class named `Fruit` and set its `name` property.
50. Write a basic `try-catch` block that anticipates and catches an arithmetic error like division by zero.

4. Analyzing (Draw connections among ideas)

51. Analyze the functional difference between a `for` loop and a `while` loop to determine when to use each.
52. Compare the GET and POST HTTP methods strictly in terms of data visibility and security.
53. In Object-Oriented Programming, how does a class fundamentally relate to an object?
54. Analyze the inline comment in `$x = 5 /* + 15 */ + 5;` and determine the final executed value of `$x`.
55. Contrast the approach of procedural programming with that of object-oriented programming in PHP.
56. Examine the role of the `$_FILES` superglobal array during the file upload process; what specific information does it capture?
57. Analyze why the release of PHP 5 in 2004 significantly improved performance for large-scale web applications.
58. Break down how page watermarking operates to secure the intellectual property of a PHP web application.

59. Analyze the execution flow of a `goto` statement and explain how it differs from normal sequential code execution.
60. Break down the output of `strpos("Hello world!", "world");` and explain why the result is exactly 6.
61. Compare the `fread()` function with `file_get_contents()`. Which is structurally simpler for reading plain text files?
62. Outline the four distinct CRUD operations and map them to their corresponding database management actions.
63. Analyze how an infinite loop occurs during runtime and identify the common logical error that causes it.
64. Break down the individual components of the XAMPP software package acronym.
65. Examine the structural benefits of using a `foreach` loop specifically for iterating through arrays.
66. Compare the strict definition of a syntax error against a runtime error in PHP.

5. Evaluating (Justify a stand or decision)

67. Evaluate the security risks associated with using the GET method for a login form on the Intego Corporates website.
68. Assess whether sessions or cookies are more technically appropriate for storing highly sensitive user authentication data, and justify your stance.
69. Critique the "one-way" nature of the `md5()` function. Why is it an effective tool for safely tracking unique visitor IPs?
70. A developer executes `unlink()` without verifying the target file. Evaluate the permanent consequences of this action.
71. Evaluate the long-term maintenance advantages of Object-Oriented Programming (OOP) over procedural programming for a growing startup project.
72. Assess the absolute necessity of the `break` statement inside a `switch` case block.
73. Judge the operational appropriateness of using a static web server versus a dynamic web server for a CMS platform like WordPress.
74. Evaluate the mathematical logic in the code `$total = $price + $discount;` when calculating a final sale price. What is the flaw?

75. Determine the most logical loop type (For, While, Do-While) for continuously fetching database rows when the total count is unknown.
76. Evaluate the critical importance of the `Content-Disposition: attachment` HTTP header when forcing a browser to download a file.
77. Justify the software engineering concepts of data abstraction and encapsulation when designing modular classes.
78. Critique the readability and efficiency of deeply nested `if-else` statements compared to a streamlined `switch` statement for numerous conditions.
79. Assess the effectiveness of compiling PHP scripts into Bytecode (OpCache) to protect commercial intellectual property.
80. Evaluate a scenario where a child class inherits properties from a parent class. Why is this inheritance architecturally beneficial?
81. Determine the potential resource problem if `fclose()` is routinely forgotten after executing `fopen()` and `fwrite()`.
82. Evaluate the architectural improvements of PHP 8.0, such as the JIT compiler, compared to older versions like PHP 5.0.
83. Assess the validity of defining properties using the `var` keyword for compatibility with PHP 4.

6. Creating (Produce new or original work)

84. Design a complete PHP script that creates a text file named "portfolio_logs.txt", writes an initial log string, and safely closes the file stream.
85. Formulate an Object-Oriented PHP class named `Video` for the Pulse app, complete with `public` properties for title and a method to display them.
86. Construct a dynamic HTML interface integrated with PHP that accepts a student's name and submits it securely using the POST method.
87. Develop a PHP data snippet using the `mysqli` extension to retrieve and echo the `visitor_count` from a table named `site_data`.
88. Create a customized error handling block using `try-catch` that intentionally throws an `InvalidArgumentException` for negative user inputs.

89. Design a structural pagination algorithm that accurately calculates `$offset` and `$totalPages` using the ceiling function for a CMS database.
90. Develop a nested `foreach` loop architecture that successfully outputs distinct student marks across multiple academic subjects.
91. Construct a server-side file upload script that targets the "uploads/" folder and executes the `move_uploaded_file()` function.
92. Create a session-based validation script that immediately redirects unauthenticated users to a "login.php" interface.
93. Formulate a dynamic `while` loop that simulates selling virtual products until the initial stock variable reaches zero.
94. Compose a script utilizing `file_get_contents()` and `str_replace()` to locate and modify a specific configuration keyword inside a text file.
95. Design a custom parameterized function named `addNumbers()` that takes two variables and echoes their mathematical sum.
96. Generate a multidimensional array representing three different music tracks with nested keys for 'title' and 'artist' for a Spotify-vibe project layout.
97. Create a PHP initialization script that securely starts a session, assigns a generic user ID, and outputs a custom tracking watermark.
98. Construct a nested `for` loop logic sequence that iteratively prints a visually ascending pyramid pattern using the `*` character.
99. Propose an iteration snippet utilizing the `continue` jumping control statement to loop through 10 objects and intentionally skip over specific matched numbers.
100. Design an object-oriented class template for Database Connections that strictly encapsulates credentials (host, username, password) as `private` properties.