Name: _____ **Date:** _____

Answer Key: Glitch in the Matrix: High School Logic and Flow Control

Evaluate complex algorithm design through recursive thinking, Boolean logic gates, and memory management scenarios in this high-level programming assessment.

---

1. A developer is optimizing a search algorithm that needs to traverse a hierarchical file system. Why might they choose a recursive function over a standard iterative loop for this task?

Answer: B) Recursive structures naturally mirror the self-similar nodes of a tree.

Recursion is mathematically suited for tree-like data structures (like folders) because the logic for exploring one subfolder is identical to the logic for the root folder, leading to cleaner, more modular code.

2. In a banking application, a variable tracking a user's account balance must remain consistent across multiple threads. This concept of limiting a variable's visibility and protecting its state is known as _____.

Answer: C) Encapsulation

Encapsulation is a fundamental OOP principle that bundles data and methods, restricting direct access to prevent accidental or unauthorized mutation of sensitive values like account balances.

3. In short-circuit evaluation of a Boolean AND (&&) expression, if the first condition is false, the second condition is never evaluated.

Answer: A) True

To optimize performance, compilers use short-circuiting; if the first part of an AND statement is false, the entire expression must be false, so evaluating the second part is redundant.

4. Consider an AI pathfinding script where an NPC moves 'while (target_found == false)'. If the target is never reachable, this creates a logic error known as:

Answer: C) An infinite loop

If the condition for exiting a while loop is never met (the guard stays true), the program enters an infinite loop, which consumes CPU resources and causes the application to hang.

5. When passing a large dataset to a function, a programmer chooses to pass by _____ to avoid copying the entire data structure into memory twice.

Name: _____ **Date:** _____

Answer: B) Reference

Passing by reference passes the memory address of the data rather than a full copy, which is essential for performance and memory efficiency when dealing with massive objects.

6. Constants are variables whose values can be modified by the program during runtime as long as the data type remains the same.

Answer: B) False

Unlike standard variables, constants are immutable; once assigned, their value is fixed for the duration of the program's execution to ensure data integrity.

7. In a complex 'switch' or 'case' statement, what occurs if a 'break' command is omitted after a matching condition?

Answer: B) The code 'falls through' and executes subsequent cases.

Without a 'break' statement, the execution continues into the next case labels regardless of whether they match the condition, a behavior known as fall-through.

8. An array with a fixed size of 10 elements has valid indices ranging from _____ to 9.

Answer: C) 0

Most modern programming languages use zero-based indexing, meaning the first element of any collection is at position 0, not 1.

9. Which of the following scenarios best demonstrates the need for a nested loop structure?

Answer: C) Iterating through every pixel in a 2D image to apply a filter.

A 2D grid (like pixels in an image) requires a nested loop: an outer loop for the rows (y-axis) and an inner loop for the columns (x-axis) within each row.

10. A global variable is accessible only within the specific function where it was first declared.

Answer: B) False

The description provided is for 'local scope.' Global variables are declared outside of functions and are accessible by any part of the program, which can often lead to debugging challenges.