Beginner-Level Coding Interview Preparation: Problem-Solving Focused Syllabus

From Fundamentals to Confidence: A 5-Day Coding Interview Bootcamp

Day 1: Introduction & Problem-Solving Fundamentals

- **Topics:**
    - Importance of coding interviews and core problem-solving strategies.
    - Approaches to breaking down problems, understanding requirements, and developing solutions step-by-step.
    - Practice platforms (e.g., CodeKata) and tips for writing clean, efficient code.
    - **Hands-on Tasks**: Simple exercises in logic and basic algorithm design, such as finding maximum or minimum values in a list, or basic arithmetic and logical operations.
- **Goal:** Build foundational problem-solving skills and instill confidence in tackling a variety of problems.

Day 2: Core String and Integer Manipulations

- **Topics:**
    - String reversal, checking for palindromes, integer reversal, maximum character occurrence in a string, FizzBuzz problem.
    - **Hands-on Tasks**: Practice transforming and analyzing strings and numbers in ways commonly tested in interviews, such as:

      Building functions to check palindrome status, reverse integers, or count character occurrences in strings.

- **Goal:** Strengthen understanding of common string and integer manipulations, essential for foundational coding challenges.

Day 3: Arrays and Basic Object Manipulations

- **Topics:**
    - Array chunking, detecting anagrams, sentence capitalization.
    - **Hands-on Tasks:**
        - Exercises in dividing and restructuring arrays, verifying anagram relationships, and formatting sentences.
        - Practice with problems involving basic data structures (arrays and simple objects), such as flattening arrays or grouping items based on criteria.
- **Goal:** Develop familiarity with arrays and objects, focusing on techniques for manipulating and restructuring data.

Day 4: Patterns, Matrix Manipulation, and Logical Thinking

- **Topics:**
    - Visualizing patterns with loops (e.g., printing steps, building pyramids).
    - Simple matrix manipulations (e.g., creating spirals, transposing matrices).
    - **Hands-on Tasks:**
        - Practice creating and navigating simple matrices, building pyramid patterns, and developing solutions that involve iteration within grids or structured data.
        - Implement a variety of loop-based problems to reinforce pattern recognition and visualization skills.
- **Goal:** Improve logical thinking and problem visualization through pattern-based and matrix problems, which test a candidate's ability to work with nested structures.

Day 5: Complexity Analysis & Efficient Problem Solving

- **Topics:**
    - Introduction to Big-O notation, understanding runtime complexity, and applying complexity analysis.
    - Simple examples of efficiency analysis with the Fibonacci series and sorting techniques.
    - **Hands-on Tasks:**
        - Practice assessing the efficiency of various code snippets and problem solutions.
        - Apply optimization techniques to common problems, such as sorting arrays or finding the nth Fibonacci number.
- **Goal:** Equip students with the skills to analyze and improve code performance, preparing them to address efficiency-related questions during interviews.