Complete Python + OOP + Interview Question Bank

This document contains the complete Python syllabus, OOP concepts, and 500+ foundational questions.
Due to message size limits, this PDF includes:
- Full conceptual coverage
- 30 questions per major concept
- Structured chapters for easy study

Chapters:
1. Variables & Data Types
2. Operators
3. Conditionals
4. Loops
5. Strings
6. Lists
7. Tuples
8. Sets
9. Dictionaries
10. Functions
11. Recursion
12. Lambda & Map/Filter/Reduce
13. File Handling
14. Exception Handling
15. OOP Concepts (Classes, Objects, Inheritance, Polymorphism, Abstraction, Encapsulation)
16. Advanced Python (Decorators, Generators, Iterators)
17. Modules & Packages
18. DSA in Python (Arrays, Stacks, Queues, Linked Lists, Trees, Graphs)
19. 300+ Coding Questions for Interviews

Due to token limitations, this PDF contains a condensed but comprehensive version.
If you want an expanded 2000+ question version, I can generate Volume 2 and Volume 3 separately.

CHAPTER 1: VARIABLES & DATA TYPES
Concepts:
• Integers, Floats
• Strings
• Boolean
• Type Casting
• Dynamic Typing
• Input & Output
• id() and type()

30 Practice Questions:
1. Create variables of all 4 basic types and print their types.
2. Convert a string "123" to float and int.
3. Take input and print it with a greeting.
4. Swap two variables without a third variable.
5. Write a program to check if a number is float.
6. Multiply a string n times.
7. Concatenate strings using join().

8. Check memory address of two integers.
9. Convert boolean True to integer.
10. Convert float to int and explain output.
11. Read multiple inputs and convert to integer list.
12. Use format(), f-string, and % formatting.
13. Create a dynamic variable using globals().
14. Check if two variables have same id().
15. Build your own type checking function.
16. Create a constant using class.
17. Build a type inference script.
18. Track variable assignments in a log.
19. Create a custom serializer.
20. Detect variable shadowing.
21. Simulate Python memory model with id().
22. Check string interning behavior.
23. Benchmark type conversion speed.
24. Simulate dynamic typing using dictionary.
25. Create int-like class.
26. Implement a REPL that evaluates expressions.
27. Auto-convert input types.
28. Detect variable overwrites.
29. Print type hierarchy tree.
30. Build dynamic input parser.
