32-Day Python Logic Mastery Plan (Oct 15 – Nov 15, 2025)

Objective

Build on your existing Python foundation by targeting weak areas, practicing core algorithms, and completing project-based assessments to achieve 75%+ accuracy by Nov 15.

Week 1: Fix Your Failed Questions (15–21 Oct)

15 Oct (Today):

- Q3: Sum of even numbers only (1–100)
- Practice: Sum odd numbers (1–50)
- Practice: Print multiples of 5 (10–100)

16 Oct:

- Q4: Count vowels in a string
- Q7: Reverse string without slicing
- Practice: Count consonants

17 Oct:

- Q8: Count character occurrences without `.count()`
- Practice: Most frequent character
- Practice: Count digits vs letters

18 Oct:

- Q11: Fix `is_prime()` logic
- Test: 2, 7, 15, 29, 100
- Practice: Write `is_odd()`, `is_even()`

19 Oct:

- Q12: Factorial using loop (no recursion)
- Test: 0, 1, 5, 10
- Practice: Sum of 1 to n function

20 Oct:

- Q6: 2nd largest without `sort()/max()`
- Practice: 3rd largest

21 Oct:

- Q13: First non-repeating character (fix logic)
- Q14: Anagram check (frequency method)
- Practice: Word frequency in sentence

Week 2: Core Algorithms (22–28 Oct)

22 Oct: Bubble sort manual; practice selection sort

23 Oct: Q15: Perfect number check; test 6, 28, 12; find all perfect numbers ≤1000

24 Oct: List comprehension and filtering (even, odd, multiples of 3/5)

25 Oct: Nested loops patterns (triangle, inverted, pyramids, Pascal’s triangle)

26 Oct: Advanced string ops (palindrome, remove duplicates, substring check)

27 Oct: Functions with lists (even numbers, primes in range, list intersection)

28 Oct: Mini Test 1 (10 questions, 60 min, target 7/10)

Week 3: Real Logic Building (29 Oct – 4 Nov)

29 Oct: Review Mini Test 1 mistakes

30 Oct: Two-pointer techniques (reverse array, palindrome check, merge sorted arrays)

31 Oct: Advanced dictionary (word frequency, group anagrams, first non-repeat char)

1 Nov: File handling (write/read/append)

2 Nov: Exception handling (zero division, invalid input, index errors)

3 Nov: Recursion basics (factorial, Fibonacci, digit sum)

4 Nov: Mini Test 2 (12 questions, 75 min, target 9/12)

Week 4: Project + Final Push (5–11 Nov)

5 Nov: Review Mini Test 2

6 Nov: Choose mini project (number guessing, contact book, expense tracker)

7–9 Nov: Build project in 3 phases

10 Nov: Code review, comments, README, edge-case testing

11 Nov: Speed practice (5 easy in 30 min, 5 medium in 45 min)

Week 5: Final Assessment (12–15 Nov)

12 Nov: Mock test prep (patterns, strings, functions, dicts, lists)

13 Nov: Full mock test (20 questions, 90 min, target 15/20)

14 Nov: Review mock test, fix mistakes, note missed patterns

15 Nov: Buffer day: light practice, favorite problems, plan next phase

Daily Routine

- **Mon–Sat:**
  - 7:00–8:30 AM: Python practice (1.5 h)
  - 3:30–5:30 PM: Afternoon session (2 h)
  - 8:00–9:00 PM: Review & journal (1 h)
- **Sunday:**
  - 9:00 AM–12:00 PM: Extended practice (3 h)
  - 4:00–6:00 PM: Project work (2 h)

Total weekly practice: ~27 h

Success Metrics by Nov 15

- 75%+ accuracy on final mock test
- Solve loop/string/function problems in ≤15 min
- Clean, well-named code handling edge cases
- Completed mini project with full features

Stay consistent, track your progress daily, and review mistakes promptly.