Applied Python: From Fundamentals to Portfolio

Instructor Guide - v1.0 (regenerated) - 2025-12-18

Delivery Modes

- Self-paced: learners complete stages independently; review via rubric and code review checklists.
- Instructor-led: 1-2 synchronous sessions per week; live coding + lab time.
- Hybrid: weekly stand-up + asynchronous work; GitHub PRs for feedback.

Recommended Weekly Cadence

Week	Focus	Key Activities	Evidence
1	Stage 1	Setup, fundamentals, utility CLI	Repo + demo run
2-3	Stage 2	Data structures + exercises	Challenge solutions + toolkit
4	Stage 3	Files/logging/ argparse	CLI tool + sample report
5-6	Stage 4	API client + integration	API report + caching
7	Stage 5	Tests + packaging + CI	Tests passing in CI + package build
8-10	Stage 6	Capstone + release	Release tag + docs + demo transcript

Code Review Checklist

- Correctness: outputs match examples; handles invalid input gracefully.
- Security: no secrets in repo; uses timeouts for network; validates file paths.
- Maintainability: functions are small; names are descriptive; duplication minimized.
- Observability: meaningful logging; errors include context; no noisy prints.
- Testing: deterministic tests; mocks for network/I/O; tests cover core logic.

Suggested Grading Notes

Use the rubric in the Student Guide. For projects with ambiguous scope, grade on:

- Quality of assumptions documented in README
- Quality of edge-case handling
- Evidence of incremental development (commit history)
- Professional release hygiene (versioning, changelog, tags)