10-Week Python Mastery Plan

I'll design it topic-wise, with Easy → Medium → Hard assignments. Each week ends with a **mini-project** to combine the week's topics.

Week 1: Core Foundations

- **Topics:** Syntax, variables, control flow, strings, collections.
- **Assignments:**
  - Easy: Year-100 problem, frequency counter.
  - Medium: Calculator, group_by.
  - Hard: REPL with variables.
- **Mini Project:** Build a text analyzer that reads a file, counts words, finds top-k frequent words, and exports JSON.

Week 2: Functions & OOP

- **Topics:** Functions, closures, decorators, OOP basics.
- **Assignments:**
  - Decorators, memoization, function dispatch.
  - Shapes hierarchy, plugin system.
- **Mini Project:** Create a **"Math Toolkit"** library with functions, decorators for timing/memoization, and class-based geometry objects.

Week 3: File I/O & Exceptions

- **Topics:** File formats (txt/csv/json), error handling.
- **Assignments:**
  - CSV → JSON, CSV diff tool, streaming JSONL processor.
  - Retry decorator, structured error logs.
- **Mini Project: Log Analyzer** that parses multiple log files, handles corrupt entries, retries parsing, and outputs summaries.

Week 4: Iterators, Generators, FP

- **Topics:** Iterators, generators, itertools, functional programming.
- **Assignments:**
  - Prime generator, sliding window, producer-consumer pipeline.

    - map/filter/reduce reimplementation.
- **Mini Project:** Build a **streaming pipeline** that reads a large file in chunks, processes in windows, and computes rolling averages.

Week 5: Testing & Debugging

- **Topics:** Unit testing, pytest, debugging, logging.
- **Assignments:**
    - Unit tests for modules, fixtures & mocks, property-based tests.
    - Logging with rotation, diagnostics toolkit.
- **Mini Project:** Convert one earlier mini project (e.g., Math Toolkit) into a **fully tested & logged package** with CI pipeline (GitHub Actions).

Week 6: Algorithms & Data Structures

- **Topics:** Searching, sorting, trees, graphs, heaps, tries.
- **Assignments:**
    - Binary search, merge/quick sort, Kth smallest.
    - BST, Dijkstra, Trie.
- **Mini Project:** Implement a **search engine prototype**:
    - Build inverted index (dict of words → doc IDs).
    - Support prefix queries (Trie).
    - Rank results (TF-IDF + heap).

Week 7: Concurrency & Networking

- **Topics:** threading, multiprocessing, asyncio, sockets.
- **Assignments:**
    - Threading demo, worker pool, async web scraper.
    - TCP echo server/client, mini HTTP server.
- **Mini Project:** Build a **distributed web scraper** using asyncio + sockets. Store results in JSON/CSV.

Week 8: Web Dev + Databases

- **Topics:** Flask/FastAPI, SQLite/Postgres, ORMs.
- **Assignments:**
    - Simple Flask app, REST API, authenticated app.
    - SQLAlchemy models, multi-tenant schema.

- **Mini Project:** Create a **Task Manager App** with FastAPI + SQLAlchemy (CRUD, JWT auth, background workers).

Week 9: Performance, Packaging, Deployment

- **Topics:** Profiling, optimization, packaging, CI/CD.
- **Assignments:**
  - Profile bottlenecks, package with pyproject.toml.
  - GitHub Actions pipeline.
- **Mini Project:** Take Task Manager App, **dockerize it** and deploy locally with Postgres.

Week 10: Data & ML Basics

- **Topics:** Pandas, visualization, scikit-learn.
- **Assignments:**
  - CSV → DataFrame, cleaning, streaming pipeline.
  - Linear regression, classification pipeline.
- **Mini Project: Movie recommender system** using Pandas + scikit-learn (content-based filtering).

🎯 Capstone Projects — Combining Topics

Here's the **permutation/combination style mix** of topics → projects. Each one uses multiple areas so you're forced to think holistically.

1. Smart CLI File Organizer

- Topics: Strings, File I/O, OOP, Logging, Testing.
- Features: Organize files into folders by type, log actions, provide CLI interface.

2. Chatroom Application

- Topics: Sockets, Concurrency, Exceptions, Logging.

- Features: Multi-client chat app, broadcast messages, logs with retries.

3. Financial Portfolio Analyzer

- Topics: File I/O, Pandas, Plotting, Error handling.
- Features: Load CSV stock prices, compute metrics, generate plots & PDF report.

4. Recipe Recommendation API

- Topics: FastAPI, SQLAlchemy, Testing, ML Basics.
- Features: Store recipes, query by ingredients, recommend similar recipes (cosine similarity).

5. Search Engine for E-Books

- Topics: Strings, File I/O, Data Structures (Trie, Heap), Generators.
- Features: Index `.txt` files, support prefix queries, rank results.

6. Distributed Web Scraper with Dashboard

- Topics: asyncio, aiohttp, FastAPI, Databases, Pandas.
- Features: Scrape sites concurrently, store in DB, provide REST API + simple dashboard.

7. Job Queue System (Mini Celery)

- Topics: OOP, Concurrency, Databases, Logging, Testing.
- Features: Submit jobs (tasks), workers process in background, track status in DB.

8. IoT Sensor Data Collector

- Topics: Sockets, Asyncio, Pandas, Visualization, CI/CD.
- Features: Collect data from simulated IoT devices, store, plot, trigger alerts.

---

9. E-Commerce Backend Prototype

- Topics: FastAPI, SQLAlchemy, Testing, JWT Auth, Logging.
- Features: Products, carts, checkout simulation, logging & monitoring.

---

10. Knowledge Graph Question Answering

- Topics: Strings, Graph DS, Pandas, ML Basics.
- Features: Build knowledge graph from CSV, allow queries, simple ML classifier for answering.