📅 Day 1–2: Python Refresher

- **Topics:**
    - Data types and structures (lists, dicts, sets, tuples)
    - Functions and decorators
    - List comprehensions, generators
    - Exception handling
    - Modules and packages

Questions:

- How can you handle exceptions gracefully?
- When would you use a generator vs a list comprehension?

📅 Day 3–4: File Handling and Automation

- **Topics:**
    - Reading/Writing files (`open()`, `with`, `os`, `shutil`)
    - CSV/JSON/YAML handling (`csv`, `json`, `yaml` libraries)

Questions:

- Write a Python script to parse YAML configuration files.
- How would you automate cleaning temporary files older than 7 days?

📅 Day 5–6: Virtual Environments & Dependency Management

- **Topics:**
  - `venv` and `virtualenv`
  - `pip` and dependency management (`requirements.txt`)
  - `pipenv` basics

Questions:

- How can you replicate your development environment for your team?
- Why use virtual environments in DevOps?

📅 Day 7–8: Python Scripting for System Administration

- **Topics:**
  - Using `subprocess` and `os.system`
  - Process management (`psutil` library)
  - Interacting with OS (Linux) commands (cron, systemctl)

Questions:

- Write a script that monitors CPU and memory usage.
- How can you manage system services via Python scripts?

📅 Day 9–10: Python Networking & APIs

- **Topics:**
  - HTTP requests (`requests` library)

    - RESTful API interactions
    - Handling API responses (JSON parsing)

Questions:

- Automate retrieving data from a public API and store in JSON.
- How can you handle rate limits and authentication when using APIs?

📅 Day 11–12: Database Integration

- **Topics:**
    - SQLite, MySQL, PostgreSQL basics (SQLAlchemy)
    - Connection pooling
    - CRUD operations via Python

Questions:

- Write Python code to interact with a database using SQLAlchemy.
- How would you handle database connection errors?

📅 Day 13–14: Infrastructure as Code with Python

- **Topics:**
    - Introduction to Terraform and AWS CDK
    - Automating AWS/GCP/Azure resources using Python scripts (Boto3)

Questions:

- Automate the creation of an EC2 instance using Boto3.
- How can Python scripts simplify cloud resource management?

📅 Day 15–16: Containers and Docker with Python

- **Topics:**
  - Docker basics (Dockerfile, images, containers)
  - Python web app Dockerization
  - Docker Compose basics

Questions:

- How do you optimize a Python Docker image?
- Write a Dockerfile for a Python web app.

📅 Day 17–18: CI/CD Automation (Jenkins/GitHub Actions) with Python

- **Topics:**
  - Python test automation (pytest, unittest)
  - Python scripts in Jenkins pipeline
  - GitHub Actions workflows for Python apps

Questions:

- Set up a CI/CD pipeline for a Python app using GitHub Actions.
- How can Python scripts be used to automate test reporting?

📅 Day 19–20: Monitoring, Logging, and Alerting

- **Topics:**
  - Logging best practices (`logging` module)
  - Monitoring with Prometheus and Grafana using Python exporters
  - Error reporting and alerting with Python scripts

Questions:

- How can you implement effective logging in a Python microservice?
- Set up Python-based alerting if a system metric crosses a threshold.

📅 Day 21: Security Automation & Compliance

- **Topics:**
  - Security best practices (OWASP guidelines)
  - Python scripts for security scans (Bandit, safety)
  - Automating vulnerability checks

Questions:

- Automate vulnerability scanning in your Python app.
- How to integrate Bandit or similar tools into a CI pipeline?

📅 Day 22: Configuration Management (Ansible & Python)

- **Topics:**
    - Basics of Ansible with Python scripts
    - Creating custom Ansible modules with Python

Questions:

- Write a simple Ansible Python module.
- How to integrate Ansible automation into larger infrastructure projects?

📅 Day 23: Project & Portfolio Development

- **Topics:**
    - Build and deploy a complete Python-based automation project integrating concepts learned above (Docker, CI/CD, APIs, monitoring, cloud integration).

Questions:

- Demonstrate a deployed Python application using Docker, Terraform, and Jenkins/GitHub Actions.
- Document your process clearly in a GitHub repository.

🚩 Important Concepts to Master:

- Automation scripting and system administration
- Infrastructure as Code and cloud interaction (AWS/Azure/GCP)
- CI/CD pipeline automation
- Containerization and orchestration (Docker/Kubernetes basics)

- Monitoring, logging, and security automation

🎯 List of Interview-Level Questions to Answer (Key for Expertise):

Python & Automation:

1. Explain how you handle exceptions in Python scripts for production systems.
2. What are Python context managers, and how do you use them?

Cloud and IaC:
3. How can Python be used effectively in infrastructure automation (Terraform vs Boto3)?
4. Explain how you'd automate cloud provisioning and destruction in AWS.

Containers & Docker:
5. What techniques do you use to optimize Docker images for Python?
6. Explain multi-stage builds in Dockerfiles.

CI/CD Pipelines:
7. How do you automate Python app testing in a Jenkins pipeline?
8. Describe a fully automated CI/CD pipeline for Python applications.

Monitoring & Logging:
9. What logging strategies do you use in Python production apps?
10. How do you set up alerting based on Python script outputs?

Security & Compliance:
11. What security practices do you implement in your Python code?
12. How do you automate security compliance checks with Python?

Configuration Management:
13. How can Python enhance or extend Ansible automation?
14. Describe the creation and usage of custom Python modules in Ansible.

🔖 Bonus Projects (Optional):

- Create a monitoring dashboard using Prometheus and Grafana fed by Python scripts.

- Develop a Python-based CLI tool for cloud resource management.
- Automate deployment of Kubernetes resources using Python client libraries.