# Integrated Python for Network Engineers – Final Curriculum

## Course Overview

This program integrates Cisco’s Programming for Network Engineers (PRNE), NetAcad Python Essentials 1 & 2 (PCEP and PCAP readiness), and a hands-on Python mini-curriculum emphasizing best practices, network automation, and applied labs. It is designed as a 5-day intensive program with structured lectures, guided demos, and labs that progressively build toward a capstone automation project.

## Duration

5 consecutive days, 8 hours per day (40 hours total).

## Objectives

By the end of the program, participants will be able to:

* - Understand Python fundamentals: variables, types, loops, functions, modules, classes, and error handling.
* - Apply Python to parse network data (text, CSV, JSON, XML) and interface with APIs (RESTCONF, Junos REST).
* - Use libraries such as Netmiko, NAPALM, and HTTPX to interact with Cisco and Juniper devices safely.
* - Build reusable, idempotent scripts that support dry-run and validation modes.
* - Develop debugging, testing, and validation workflows for reliable automation.
* - Plan and deliver a capstone multi-vendor automation project.

## Expected Outcomes

* - Proficiency in Python syntax, data types, control flow, functions, and OOP.
* - Ability to safely connect to and configure network devices using Python libraries.
* - Practical skills in file handling, regex, JSON/XML/CSV data processing.
* - Confidence in using modules, packages, and environments to structure code.
* - Experience with concurrency, retries, and error handling in network automation.
* - Completion of a working capstone automation project with validation artifacts.

## Course Outline (5-Day Schedule)

### Day 1 — Python Foundations & Device Access

* - Orientation, goals, and environment setup
* - Python basics: variables, types, flow control
* - Functions, error handling, logging
* - Lab: inventory parsing, first scripts, Netmiko basics

### Day 2 — Data Structures, Persistence, and NAPALM

* - Lists, dicts, sets, tuples; iteration patterns
* - YAML/CSV → JSON/SQLite persistence
* - Lab: NAPALM getters for interfaces, routes, facts
* - Validation scripts and structured outputs

### Day 3 — REST APIs and Parallelism

* - HTTP basics, JSON handling, retries, pagination
* - Cisco RESTCONF, Junos REST API (with mocks/offline)
* - Lab: REST queries with validation and dry-run
* - Concurrency with ThreadPoolExecutor (timeouts, retries)

### Day 4 — Idempotent Configuration & Ansible Intro

* - Idempotence patterns: dry-run, diffs, validation
* - Lab: VLAN + interface config across Cisco & Junos
* - Optional: Ansible playbooks in check mode
* - Code review and capstone scaffolding

### Day 5 — Capstone Project & Wrap-Up

* - Integration: inventory → API query → idempotent config
* - Lab: build and demo capstone automation project
* - Post-assessment and knowledge quiz
* - Next steps and resources

## Software and Tools

* - Python 3.11+ (venv, pip, Black formatter)
* - Libraries: Netmiko, NAPALM, Nornir, HTTPX, Requests, TextFSM, Jinja2
* - Cisco DevNet Sandbox, Juniper vLabs (with offline fixture fallback)
* - Version control: GitHub/GitLab/Bitbucket/Azure DevOps
* - IDE: VS Code with WSL or Bash integration

## Target Audience

* - Network Engineers and Administrators with little or no Python experience
* - Systems Engineers moving into automation
* - Network Managers seeking foundational programming skills
* - Learners preparing for Cisco DevNet Associate, PCEP, or PCAP exams

## Labs and Assessments

* - Hands-on labs each day (device access, data parsing, REST, idempotence)
* - Daily checkpoints, quizzes, and code reviews
* - Validation scripts and structured output checks
* - Capstone project: multi-vendor automation demo
* - Optional exam prep for PCEP/PCAP