

Course Name : **PCEA CERTIFIED ENTRY LEVEL AUTOMATION SPECIALIST WITH PYTHON**
Duration : 3 Days (Physical Classroom / Virtual Live Instructor)
Skill Level : Beginner

COURSE DESCRIPTION:

Build practical Python automation skills through the PCEA – Certified Entry-Level Automation Specialist with Python certification training. This course covers automation fundamentals, command-line scripting, logging and monitoring, file and data automation, web and API automation, scheduling, notifications, and reporting.

Participants will learn how to use Python to automate simple, well-defined tasks, work with files and data, monitor automation with logs, interact with web resources and APIs, schedule tasks, send notifications, and generate basic reports.

WHAT WILL YOU LEARN?

Participants will develop the foundational knowledge and practical skills required for PCEA-30-01, including command-line automation, virtual environments, environment variables, subprocess, logging, file handling, CSV/JSON processing, web/API automation, scheduling, notifications, and reporting.

PREREQUISITE:

Foundational Python programming knowledge at PCEP level or equivalent is recommended, together with introductory exposure to practical automation concepts. No professional automation experience is required.

METHODOLOGY:

This program will be conducted with interactive lectures, PowerPoint presentations, discussions, demonstrations, and practical exercises. This course can be conducted as instructor-led (ILT) or virtual instructor-led training (VILT).

JOB SCOPE:

Upon completion of this course, candidates may pursue the following career paths:

- Junior Data Analyst
- Entry-Level Data Analyst
- Data Analytics Assistant
- Junior Python Data Analyst
- Data Reporting Analyst
- Business Data Analyst – Entry Level
- Data Operations Analyst
- Reporting & Analytics Executive
- Junior Data Specialist

MODULE 1: FUNDAMENTALS OF AUTOMATION

The Importance of Digitizing Tasks

- Identify routine and repetitive tasks suitable for automation in IT, business, and everyday contexts.
- Distinguish rule-based, time-consuming tasks from tasks requiring creativity or judgment.
- Benefits and Limitations of Automation
- Explain consistency, accuracy, speed, scalability, cost savings, time savings, and productivity benefits.
- Recognize setup costs, script errors, system dependencies, and maintenance overhead.

Levels of Automation

- Differentiate scripting, process automation, and orchestration.

Measuring the Value of Automation

- Apply basic ROI reasoning using time saved, error reduction, and cost savings.

Python as a Tool for Automation

- Explain Python's readability, cross-platform compatibility, standard library, and community support.
- Identify automation libraries including os, shutil, subprocess, logging, requests, and schedule.

MODULE 2: BASIC COMMAND-LINE AUTOMATION WITH PYTHON

Running Python Scripts from the Command Line

- Run Python scripts from terminal/command prompt and diagnose common path or interpreter errors.
- Pass runtime arguments using sys.argv and use arguments to control script behavior.
- Create, activate, and deactivate virtual environments to isolate dependencies and improve portability.

Script Configuration and Execution

- Use shebang lines for Unix/Linux/macOS scripts and make scripts executable with chmod +x.
- Redirect stdout and stderr and analyze output to distinguish successful runs from errors.
- Work with environment variables such as PATH, HOME, and USERNAME using os.environ.

Integration with System Tools

- Understand Windows Task Scheduler and cron for OS-level scheduling.
- Use subprocess to execute external commands, capture output, and inspect return codes.
- Identify command-line automation use cases such as backups, batch renaming, and log cleanup.

Integration with System Tools

- Compare OS-level scheduling with in-script scheduling for appropriate automation scenarios.
- Combine Python scripts with external commands and system tools to reduce repetitive manual work.

MODULE 3: LOGGING AND MONITORING ESSENTIALS

Understanding the Role of Logging

- Explain logging for debugging, monitoring, auditing, and traceability in automation.
- Explain the limitations of print() compared with structured logging.
- Recognize why passwords, API keys, and other sensitive information should never be logged.
- Configuring Logging in Python
- Configure basic logging with logging.basicConfig and write messages to console and files.
- Use DEBUG, INFO, WARNING, ERROR, and CRITICAL appropriately.

Log Levels and Formats

- Configure timestamps, log levels, and message details with custom formats.
- Use consistent log levels to organize and analyze automation logs.
- Logging File Operations
- Record file reads/writes and the number of lines or records processed.

- Compare logs of successful operations with failed attempts.

Monitoring Automation Tasks

- Record script start/end times, processed files or records, and execution errors.
- Interpret logs to troubleshoot failed backups, API timeouts, and other automation issues.

MODULE 4: BASIC FILE AND DATA AUTOMATION

File and Folder Operations

- Use `os` to list files, verify paths, create/delete directories, and perform housekeeping tasks.
- Read, write, and append text files safely, including UTF-8 encoded files.
- Use `shutil` to copy, move, rename, and back up files and directories.
- Handle missing files and permission errors with `try/except` and logging.

CSV and JSON Processing

- Differentiate CSV and JSON use cases and select an appropriate format.
- Read/write CSV with `csv.reader()`, `DictReader()`, `csv.writer()`, and `DictWriter()`.
- Parse/generate JSON with `json.load()`, `loads()`, `dump()`, and `dumps()`.

Professional Practices in Data Automation

- Use context managers, safeguards, and error handling for reliable file operations.
- Detect empty/corrupted files and prevent automation failures or data loss.
- Apply safe file naming, version control, and logging for traceability.
- Protect sensitive information and apply privacy, security, and responsible automation practices.
- Recognize privacy risks of temporary files and the need for appropriate safeguards.

MODULE 5: BASIC WEB AND API AUTOMATION

Introduction to Web Automation

- Differentiate web scraping from API-based data access and choose an appropriate approach.
- Recognize ethical and legal considerations including `robots.txt`, rate limits, throttling, and responsible requests.
- Using requests for Web Content
- Fetch web content with `requests.get()` and interpret common HTTP status codes such as 200, 404, and 500.
- Apply basic error handling to detect failed requests.

Parsing JSON and HTML

- Parse JSON responses with `response.json()` and extract required values.
- Use BeautifulSoup to extract titles, links, paragraphs, and other simple HTML elements.
- Log request success/failure and the number of elements processed.

Working with REST APIs

- Explain GET and POST methods, JSON responses, API keys, authentication, and rate limits.
- Differentiate public and private APIs and recognize common access restrictions.
- Fetch data from simple APIs, parse JSON, and store results in CSV or JSON files.

MODULE 6: SCHEDULING, NOTIFICATIONS, AND REPORTING

Scheduling Basics

- Explain why repetitive tasks should be scheduled and distinguish manual execution from scheduled runs.
- Use Python's schedule module for fixed-interval, daily, and weekly tasks.
- System-Level Scheduling
- Describe cron jobs and Windows Task Scheduler and compare them with Python-based scheduling.
- Recognize risks such as misconfigured jobs, overlapping executions, and missed triggers.

Notifications and Alerts

- Send basic email notifications using smtplib and handle email credentials securely.
- Identify desktop notification options such as plyer and appropriate notification use cases.
- Reporting in Automation
- Generate plain-text and basic HTML reports with headings, tables, lists, and timestamps.
- Save reports to designated directories for organization and retrieval.
- Apply logging to scheduled and reporting workflows for traceability and troubleshooting.

PCEA-30-01 EXAM PREPARATION

- Review the six PCEA examination blocks and their key competencies.
- Practice applying Python to command-line, file/data, logging, web/API, scheduling, and reporting tasks.
- Review safe file handling, privacy, security, ethical web automation, and responsible use of automation.

MODULE 5: BASIC WEB AND API AUTOMATION

- Block 1: Fundamentals of Automation – 6 items (13%).
- Block 2: Basic Command-Line Automation with Python – 9 items (19.5%).
- Block 3: Logging and Monitoring Essentials – 7 items (15%).
- Block 4: Basic File and Data Automation – 8 items (17.5%).
- Block 5: Basic Web and API Automation – 8 items (17.5%).
- Block 6: Scheduling, Notifications, and Reporting – 8 items (17.5%).
- Total: 46 items; passing requirement is a cumulative average score of at least 75% across all exam blocks.

CONCLUSION

- Q&A
- Useful References and Documentation
- Feedback