Four-Day Course Outline for PCPP2 Certification

Day 1: Python Package Development and Testing

Learning Outcomes:

- Understand how to create and distribute Python packages.
- Learn to utilize pip for package installation and management.
- Gain skills to write and execute unit tests using unittest and pytest.

Topics Covered:

- **Creating and Distributing Packages**:
    - Directory structure for Python packages.
    - Writing setup.py files for packaging.
    - Sharing and installing packages.
    - Package documentation and licensing essentials.
- **Testing Principles**:
    - Overview of unit testing frameworks.
    - Writing and running tests using unittest.
    - Advanced testing with pytest.

Activities:

- Hands-on: Create a Python package and publish it locally.
- Lab: Write test cases for a sample Python module using unittest and pytest.

Day 2: Design Patterns in Python

Learning Outcomes:

- Develop a deep understanding of object-oriented design principles.
- Apply various design patterns to solve programming challenges.

Topics Covered:

- Overview of Design Patterns.
- Key Patterns:
    - Singleton, Factory, Façade, Proxy, Observer, Command, Template Method, MVC, and State.
- Use cases for each design pattern.

Activities:

- Group exercise: Implementing the Singleton and Factory patterns.
- Case study: Applying the Observer pattern in a real-world scenario.

Day 3: Interprocess Communication and Network Programming

Learning Outcomes:

- Master process-based and thread-based parallelism.
- Develop applications utilizing Python networking features.

Topics Covered:

- **Interprocess Communication**:
    - multiprocessing: Process-based parallelism.
    - threading: Thread-based parallelism.
    - Synchronization tools: queue, socket, and mmap.
- **Python Network Programming**:
    - Socket programming basics.
    - Server vs. client socket methods.
    - Understanding peer-to-peer networking.

Activities:

- Lab: Build a simple client-server communication model using Python sockets.
- Group activity: Implement a parallel processing task using multiprocessing.

Day 4: Database Interaction and Capstone Project

Learning Outcomes:

- Understand relational database principles.
- Develop CRUD applications using Python and MySQL.
- Integrate learned concepts in a capstone project.

Topics Covered:

- Relational Database Basics:
    - Overview of relational databases and SQL syntax.
    - Comparison: MySQL vs. other database systems.
- CRUD Applications:
    - Connecting to MySQL databases with Python.
    - Performing Create, Read, Update, and Delete operations.

Activities:

- Lab: Build a CRUD application with Python and MySQL.
- Capstone Project: Design and implement a multi-featured Python application that integrates packaging, design patterns, interprocess communication, and database access.

Course Deliverables

- Hands-on practice assignments and solutions.
- Capstone project showcasing real-world applications.
- Mock exams aligned with PCPP2 objectives.