# Certified Professional Python Developer (CPPD)

**Syllabus: CPPD-3.11**


## Introduction

The Certified Professional Python Developer Exam (CPPD) is a high-standard certification tailored for professionals aspiring to validate and showcase their advanced mastery in Python programming, testing candidates on an extensive array of advanced concepts and practical implementations within the Python ecosystem.

Acquiring the CPPD credential signifies a remarkable level of proficiency and a notable edge in the competitive landscape of Python development.

## Exam Syllabus

**Module 1: Advanced Function Concepts:**

- Grasping and utilizing advanced function attributes, closures, and decorators for creating maintainable and efficient code structures.
- Advanced lambda functions and functional programming concepts for effective code modularization and reuse.

**Module 2: Exception Handling:**

- Proficiency in identifying, handling, and propagating exceptions using try, except, finally, and raise clauses.
- Custom exception creation for effective error handling tailored to specific program requirements.

**Module 3: Comprehensive Expressions:**

- Mastery in crafting list, set, and dictionary comprehensions for streamlined data processing.
- Employing nested comprehensions and conditionals for solving intricate data manipulation tasks.

**Module 4: Regular Expressions (Regex):**

- Development and interpretation of complex regular expressions for sophisticated text pattern recognition and manipulation.
- Utilization of Python's re module for performing various regex operations like search, match, findall, sub, and compile.

**Module 5: File Handling:**

- Advanced techniques for reading from and writing to files, managing file contexts, and handling file exceptions.
- Employing different file formats and libraries for efficient data storage, retrieval, and processing.

**Module 6: Object-Oriented Programming (OOP):**

- Profound understanding and application of advanced OOP principles including inheritance, encapsulation, and polymorphism.
- Designing, implementing, and testing complex system architectures adhering to OOP paradigms.