

# Advanced Programming Techniques with Python

## Course Specifications

Course Number: ELK94-022\_rev1.3

Course Length: 3 days

## Course Description

Python® continues to be a popular programming language, perhaps owing to its easy learning curve, small code footprint, and versatility for business, web, and scientific uses. Python is useful for developing custom software tools, applications, web services, and cloud applications. In this course, you'll build upon your basic Python skills, learning more advanced topics such as object-oriented programming patterns, development of graphical user interfaces, data management, creating web service-connected apps, performing data science tasks, unit testing, and creating and installing packages and executable applications.

**Course Objective:** In this course, you will expand your Python proficiencies.

You will:

- Select an object-oriented programming approach for Python applications.
- Create object-oriented Python applications.
- Create a desktop application.
- Create data-driven applications.
- Create and secure web service-connected applications.
- Program Python for data science.
- Implement unit testing and exception handling.
- Package an application for distribution.

**Target Student:** This course is designed for existing Python programmers who have at least one year of Python experience and who want to expand their programming proficiency in Python 3.

**Prerequisites:** To ensure your success in this course, you should have experience with object-oriented programming and Python 2.x or 3.x. You can obtain this level of skills and knowledge by taking the following Logical Operations courses:

- Introduction to Programming with Python

## Hardware Requirements

For this course, you will need one computer for each student and one for the instructor. Each computer will need the following minimum hardware configurations:

- 1 gigahertz (GHz) 64-bit (x64) processor.
- 4 gigabytes (GB) of Random Access Memory (RAM).
- 32 GB available storage space.
- Monitor capable of a screen resolution of at least 1,024 × 768 pixels, at least a 256-color display, and a video adapter with at least 4 MB of memory.
- Bootable DVD-ROM or USB drive.
- Keyboard and mouse or a compatible pointing device.
- Fast Ethernet (100 Mb/s) adapter or faster and cabling to connect to the classroom network.
- IP addresses that do not conflict with other portions of your network.
- Internet access (contact your local network administrator).
- (Instructor computer only) A display system to project the instructor's computer screen.

## Software Requirements

- Microsoft® Windows® 10 (64-bit).
- Python version 3.9.0 (python-3.9.0.amd64.exe, provided with the course data files).
- PyCharm Community Edition version 2020.2.3 (pycharm-community-2020.2.3.exe, provided with the course data files).
- NOTE: Python is distributed under the Python Software Foundation License (PSFL). PyCharm Community Edition is distributed under the Apache® License 2.0.
- MySQL Community Server version 8.0.22 (mysql-installer-web-community-8.0.22.0.msi, provided with the course data files).
- NOTE: MySQL Community Server is distributed under the GPL license.
- Script to set up the MySQL connector ( Setup MySQL Connector.bat, provided with the course data files).
- If necessary, software for viewing the course slides. (Instructor machine only.)

## Course Content

### Lesson 1: Selecting an Object-Oriented Programming Approach for Python Applications

- Topic A: Implement Object-Oriented Design
- Topic B: Leverage the Benefits of Object-Oriented Programming

### Lesson 2: Creating Object-Oriented Python Applications

- Topic A: Create a Class
- Topic B: Use Built-in Methods
- Topic C: Implement the Factory Design Pattern

### Lesson 3: Creating a Desktop Application

- Topic A: Design a Graphical User Interface (GUI)
- Topic B: Create Interactive Applications

**Lesson 4: Creating Data-Driven Applications**

Topic A: Connect to Data

Topic B: Store, Update, and Delete Data in a Database

**Lesson 5: Creating and Securing a Web Service-Connected App**

Topic A: Select a Network Application Protocol

Topic B: Create a RESTful Web Service

Topic C: Create a Web Service Client

Topic D: Secure Connected Applications

**Lesson 6: Programming Python for Data Science**

Topic A: Clean Data with Python

Topic B: Visualize Data with Python

Topic C: Perform Linear Regression with Machine Learning

**Lesson 7: Implementing Unit Testing and Exception Handling**

Topic A: Handle Exceptions

Topic B: Write a Unit Test

Topic C: Execute a Unit Test

**Lesson 8: Packaging an Application for Distribution**

Topic A: Create and Install a Package

Topic B: Generate Alternative Distribution Files

**Appendix A: Mapping Python Course Content to Python Institute Certification Exams**