Advanced Python
Course Outline

Introduction:

This Advanced Python course delves deep into Python concepts to enhance your skills and knowledge of the language. It covers topics such as lambda functions, advanced list comprehensions, and the Collections module. The course also explores regular expressions, working with data using SQLite, CSV, JSON, and web scraping, as well as testing and debugging techniques. Furthermore, it emphasizes object-oriented programming with classes and objects, inheritance, properties, static and class methods, and understanding decorators. Each section includes hands-on exercises to provide practical experience in applying these advanced Python concepts.

Course Duration:

2 days

Prerequisites:

Basic Python programming experience. In particular, you should be very comfortable with:

- Working with strings.
- Working with lists, tuples and dictionaries.
- Loops and conditionals.
- Writing your own functions.

Benefits:

1. **Comprehensive Coverage:** The course covers a wide range of advanced Python topics, ensuring that you have a deep understanding of some of the advanced features of the Python language.
2. **Practical Examples:** The course includes numerous exercises and examples to illustrate advanced concepts, giving you the opportunity to apply your knowledge and skills in real-world scenarios.
3. **Hands-on Learning:** Throughout the course, you will work on practical exercises that reinforce your understanding of the material and build your confidence in using advanced Python techniques.

Outline:

1. **Advanced Python Concepts**

   In this lesson, you will learn about some Python functionality and techniques that are commonly used but require a solid foundation in Python to understand.

   - Lambda Functions
   - Advanced List Comprehensions
   - Rolling Five Dice (Exercise)
   - Collections Module
   - Creating a `defaultdict` (Exercise)
   - Counters
   - Creating a Counter (Exercise)

Advanced Python
Course Outline

- Mapping and Filtering
- Mutable and Immutable Built-in Objects
- Sorting
- Converting list.sort() to sorted(iterable) (Exercise)
- Sorting Sequences of Sequences
- Creating a Dictionary from Two Sequences
- Unpacking Sequences in Function Calls
- Converting a String to a datetime.date Object (Exercise)
- Modules and Packages

After completing this module, students will be able to:

- Work with lambda functions.
- Write more advanced list comprehensions.
- Work with the collections module to create named tuples, defaultdicts, ordereddicts, counters, and deques.
- Use mapping and filtering.
- Sort sequences.
- Unpack sequences in function calls.
- Create modules and packages.

2. Regular Expressions

Regular expressions are used to do pattern matching in many programming languages, including Java, PHP, JavaScript, C, C++, and Perl. We will provide a brief introduction to regular expressions and then we'll show you how to work with them in Python.

- Regular Expression Tester
- Regular Expression Syntax
- Python's Handling of Regular Expressions
- Green Glass Door (Exercise)

After completing this module, students will be able to:

- Understand regular expressions.
- Be familiar with Python's module.

3. Working with Data

Data is stored in many different places and in many different ways. In this lesson, you'll learn about the Python modules that help you access data.

- Virtual Environment
- Relational Databases
- Passing Parameters
- SQLite
- Querying a SQLite Database (Exercise)
- SQLite Database in Memory

- Inserting File Data into a Database (Exercise)
- Drivers for Other Databases
- CSV
- Finding Data in a CSV File (Exercise)
- Creating a New CSV File
- Creating a CSV with `DictWriter` (Exercise)
- Getting Data from the Web
- HTML Scraping (Exercise)
- XML
- JSON
- JSON Home Runs (Exercise)

After completing this module, students will be able to:

- Store data in a relational database.
- Store data in a CSV file.
- Work with data from a web page.
- Work with HTML, XML, and JSON.
- Access an API.

4. Testing and Debugging

In this lesson, you will learn to test the performance and the functionality of your Python code.

- Testing for Performance
- Comparing Times to Execute (Exercise)
- The `unittest` Module
- Fixing Functions (Exercise)
- Special `unittest.TestCase` Methods

After completing this module, students will be able to:

- Learn to test the performance of different pieces of code and to create unit
- tests to test your Python code.

5. Classes and Objects

An object is something that has attributes and/or behaviors, meaning it is certain ways and does certain things. In the real world, everything could be considered an object. Some objects are tangible, like rocks, trees, tennis racquets, and tennis players. And some objects are intangible, like words, colors, tennis swings, and tennis matches. In this lesson, you will learn how to write object-oriented Python code.

- Attributes
- Behaviors
- Classes vs. Objects
- Attributes and Methods
- Adding a `roll()` Method to Die (Exercise)

Advanced Python

Course Outline

- Private Attributes
- Properties
- Properties (Exercise)
- Objects that Track their Own History
- Documenting Classes
- Documenting the Die Class (Exercise)
- Inheritance
- Extending the Die Class (Exercise)
- Extending a Class Method
- Extending the `roll()` Method (Exercise)
- Static Methods
- Class Attributes and Methods
- Abstract Classes and Methods
- Understanding Decorators