

Module 1: Object-Oriented Programming in Python

- Classes, Instances, Attributes, Methods
- Working with class and instance data
- Copying object data using *shallow* and *deep* operations
- Inheritance and Polymorphism
- Different faces of Python methods: *static* and *class* methods
- Abstract classes vs. method overloading
- Composition vs. Inheritance
- Implementing Core Syntax
- Subclassing built-ins
- Attribute Encapsulation
- Advanced techniques of creating and serving exceptions
- Serialization of Python objects using the *pickle* module
- Making Python object persistent using the *shelve* module
- Metaprogramming (function decorators, class decorators, meta classes)

Module 2: Python Enhancement Proposals

- What is PEP?
- Coding conventions
- *PEP 20* – The Zen of Python
- *PEP 8* – Style Guide for Python Code
- *PEP 257* – Docstring Conventions
- A tour of other important PEPs.

Module 3: GUI Programming

- What is GUI and where it comes from
- Constructing a GUI
- Event-driven programming
- Currently used GUI environments and toolkits
- *tkinter* module
- *pygame* module

Module 4: The Elements of Network Programming: Working with RESTful APIs

- the basic concepts of network programming
- creating sockets in Python
- how to establish and close the connection with a server
- working with JSON and XML files
- HTTP methods
- building a sample testing environment
- CRUD
- building a simple REST client
- fetch and remove data from servers
- adding new data to servers and update the already-existing data.

Module 5: File Processing

- Processing files:
 - *sqlite3* – interacting with SQLite databases

- *xml* – creating and processing XML files
- *csv* – CSV file reading and writing
- *logging* – basics logging facility for Python
- *configparser* – configuration file parser.
- Communicating with a program's environment:
 - *os* – interacting with the operating system
 - *datetime* – manipulating with dates and time
 - *io* – working with streams
 - *time* – time access and conversions.