

```
+-- Introduction to OOP
|   +-- Definition and Principles
|   +-- Importance
|   +-- Comparison with Procedural Programming
|
+-- Basic Concepts of OOP
|   +-- Class
|   |   +-- Definition
|   |   +-- Syntax
|   |   +-- Class Attributes
|   +-- Object
|   |   +-- Definition
|   |   +-- Instantiation
|   |   +-- Object Attributes
|   +-- Method
|   |   +-- Definition
|   |   +-- Instance Methods
|   |   +-- Class Methods
|   |   +-- Static Methods
|
+-- Four Pillars of OOP
|   +-- Encapsulation
|   |   +-- Definition
|   |   +-- Access Specifiers (Private, Protected, Public)
|   |   +-- Getters and Setters
|   |   +-- Example Code
|   +-- Inheritance
|   |   +-- Definition
|   |   +-- Types of Inheritance
|   |   +-- Example Code
|   +-- Polymorphism
|   |   +-- Definition
|   |   +-- Method Overloading
|   |   +-- Method Overriding
|   |   +-- Example Code
|   +-- Abstraction
|   |   +-- Definition
|   |   +-- Abstract Classes
|   |   +-- Interfaces
|   |   +-- Example Code
|
+-- Advanced Concepts
|   +-- Constructor and Destructor (__init__, __del__)
|   +-- Self Parameter
|   +-- Class Variables vs Instance Variables
|   +-- Inheritance Details
|   |   +-- super() function
|   |   +-- Method Overriding
|   |   +-- Method Resolution Order (__mro__)
|   +-- Polymorphism Details
```

```
| | +-- Duck Typing
| | +-- Operator Overloading
|
+-- Special Methods (Magic/Dunder Methods)
| +-- __str__ and __repr__
| +-- __len__, __getitem__, __setitem__
| +-- __call__, __iter__, __next__
| +-- Operator Overloading (__add__, __sub__, etc.)
|
+-- Design Patterns in OOP
| +-- Singleton Pattern
| +-- Factory Pattern
| +-- Observer Pattern
| +-- Strategy Pattern
|
+-- Real-World OOP Examples in Python
| +-- Bank Account Class
| +-- Library Management System
| +-- Employee Management System
|
+-- Best Practices in OOP
| +-- Writing clean code
| +-- SOLID principles
| +-- Using design patterns
| +-- Unit testing
```