



INDEX

1) OOP's Part – 1	1
2) OOP's Part – 2	30
3) OOP's Part – 3	58
4) OOP's Part – 4	71
5) Exception Handling	83
6) File Handling	104
7) Multi Threading	123
8) Python Database Programming	151
9) Regular Expressions & Web Scraping	166
10) Decorator Functions	180
11) Generator Functions	185
12) Assertions	190
13) Python Logging	193



🌀 OOP's Part – 3 ~~~~~ 58

⚽ Polymorphism	59
⚽ Duck Typing Philosophy of Python	59
⚽ Overloading	62
☕ Operator Overloading	
☕ Method Overloading	
☕ Constructor Overloading	
⚽ Overriding	68
☕ Method Overriding	
☕ Constructor Overriding	

🌀 OOP's Part – 4 ~~~~~ 71

⚽ Abstract Method	72
⚽ Abstract class	73
⚽ Interface	76
⚽ Concreate class vs Abstract Class vs Inteface	78
⚽ Public,Private and Protected Members	78
⚽ __str__() Method	80
⚽ Difference between str() and repr() functions	80
⚽ Small Banking Application	81

🌀 Exception Handling ~~~~~ 83

⚽ Syntax Errors	84
⚽ Runtime Errors	84
⚽ What is Exception	85
⚽ Default Exception Handing in Python	85
⚽ Python's Exception Hierarchy	86
⚽ Customized Exception Handing by using try-except	87
⚽ Control Flow in try-except	87
⚽ How to Print Exception Information	88
⚽ try with Multiple except Blocks	88
⚽ Single except Block that can handle Multiple Exceptions	90
⚽ Default except Block	90
⚽ finally Block	91
⚽ Control Flow in try-except-finally	93
⚽ Nested try-except-finally Blocks	94
⚽ Control Flow in nested try-except-finally	95



⌚ Python Logging ~~~~~ 193

⌚ Logging Levels	194
⌚ How to implement Logging	194
⌚ How to configure Log File in over writing Mode	196
⌚ How to Format Log Messages	196
⌚ How to add Timestamp in the Log Messages	197
⌚ How to Change Date and Time Format	197
⌚ How to write Python Program Exceptions to the Log File	198
⌚ Problems with Root Logger	199
⌚ Need of Our Own Customized Logger	200
⌚ Advanced logging Module Features: Logger	200
⌚ Logger with Configuration File	203
⌚ Creation of Custom Logger	205
⌚ How to Create separate Log File based on Caller	206
⌚ Advantages of Customized Logger	208