# Python 50 Repetithon

Topic Sequence: Advanced to Simple

*CoreCode Programming Academy - Python Masterclass*

This document presents the entire Python Masterclass curriculum in **reverse order** - starting from the most advanced topics and progressing down to the simplest fundamentals. This sequence is useful for experienced programmers who want to quickly locate advanced material, or for reviewing topics from complex to basic.

## Difficulty Levels

| | |
|---|---|
| **LEVEL 6** | Expert - Metaclasses, Abstract Base Classes, Decorators |
| **LEVEL 5** | Advanced - Closures, Function Factories, LEGB Scope |
| **LEVEL 4** | Upper Intermediate - Lambda, Map/Filter, List Comprehensions |
| **LEVEL 3** | Intermediate - OOP, File I/O, Data Structures |
| **LEVEL 2** | Core - Functions, Control Flow, Branching |
| **LEVEL 1** | Beginner - Variables, Data Types, Input/Output, Setup |

## LEVEL 6 - EXPERT: Metaclasses, Abstract Base Classes & Decorators

The most advanced concepts in the curriculum. These topics require a deep understanding of Python's object model, function internals, and metaprogramming capabilities.

### 1. Metaclasses & Abstract Base Classes

`DEMONSTRATION_CODE/abc_implementation.py`

Custom metaclass named 'interface' that enforces abstract method implementation. Demonstrates how Python's metaclass __call__ mechanism can be used to prevent instantiation of classes with unimplemented abstract methods. Requires understanding of type(), __call__, and the class creation process.

### 2. Function Decorators (Production-Style)

`DEMONSTRATION_CODE/logger.py, test_logger.py`

A production-quality function decorator that wraps functions to log every call with arguments, return values, and exception details to a file. Demonstrates the decorator pattern, *args/**kwargs forwarding, exception handling in wrappers, and file-based logging. Includes test suite with recursive call demonstrations.

### 3. Advanced Class Design - Vector3D

`SESSION-092/02-Vector3D.py`

A 3D vector class implementing mathematical operations: addition, subtraction, normalization, dot product, and cross product. Demonstrates operator concepts, mathematical programming, type validation in constructors, and complex method implementations with geometric algorithms.

### 4. Advanced Class Design - Triangle

`SESSION-092/01-Triangle.py`

Triangle class with perimeter and area calculations using Heron's formula. Includes constructor validation for triangle inequality, type checking, and mathematical computation methods.

### 5. Code Object Verification & Attribute Management

`SESSION-079/CODE-OBJECT-VERIFICATION-ATTRIBUTE-MANAGEMENT-EXAMPLE.pdf`

Deep analysis of Python's code objects, attribute management mechanisms, and verification techniques. Explores the internal structure of function and class objects.

## LEVEL 5 - ADVANCED: Closures, Function Factories & Scope Mastery

Advanced function behavior including closures, higher-order functions that return functions, scope resolution mastery, and complex parameter passing. Sessions 079-088.

### 6. Closures & Function Factories

`SESSION-085/01-Returning-Function-As-If-It-Were-A-Data-Object.py`
`SESSION-086/02-FunctionFactory.py`
`SESSION-085/IDLE-LOG.pdf`

Functions that create and return other functions with captured state from the enclosing scope. Demonstrates implicit state saving through closures, function factories that generate specialized functions, and treating functions as first-class data objects.

### 7. LEGB Scope Rule - Complete Mastery

`SESSION-082/03-LEGB-Scope-Rule.py`
`SESSION-082/04-UnboundLocalError.py`
`SESSION-082/05-NameError-FreeVariable.py`

Comprehensive demonstration of Python's LEGB scope resolution: Local, Enclosing, Global, Built-in. Includes analysis of UnboundLocalError scenarios, free variable behavior, NameError conditions, and how Python resolves variable references at each scope level.

### 8. global & nonlocal Statements

`SESSION-084/04-Solution-to-Scenario-1-using-global-statement.py`
`SESSION-084/07-nonlocal-demo.py`

Using global and nonlocal statements to modify variables in enclosing scopes. Demonstrates when and why these statements are needed, the problems they solve, and their interaction with nested function definitions.

### 9. Advanced Parameter Passing (*args, **kwargs)

`SESSION-087/PARAMETER-PASSING-GURU-KILLI.py`
`SESSION-087/PARAMETER-PASSING-IDLE-LOG.pdf`
`SESSION-087/REALLY-BIG-PICTURE.pdf`
`SESSION-088/Non-Default-After-Default-IDLE-LOG.pdf`

Complete parameter passing mechanisms: positional arguments, keyword arguments, default values, *args for variable positional arguments, **kwargs for variable keyword arguments, and the rules governing their ordering. Includes the 'Really Big Picture' overview document.

### 10. Nested Function Definitions

`SESSION-079/Nested-Def-Statement/01-def-under-def.py`
`SESSION-080/01-nested-def-statement-practice.py`
`SESSION-080/02-symbol-table.py`

Defining functions within functions. Covers inner function access to enclosing scope variables, symbol table analysis, and the foundation for understanding closures.

### 11. Exception Handling (try/except)

`SESSION-088 / Various session scripts`

Handling runtime errors with try/except blocks. Catching specific exception types, understanding the exception hierarchy, and using exception handling in production-style code.

## LEVEL 4 - UPPER INTERMEDIATE: Functional Programming Concepts

Functional programming patterns including lambda expressions, higher-order functions, and list comprehensions. Sessions 051-077.

### 12. Advanced List Comprehensions (Multi-Variable)

`SESSION-077/Advanced-List-Comprehension.pdf`

Multi-variable list comprehensions with nested iterations, conditional filters, and complex expressions. Extends single-variable comprehensions to handle multiple iterables and compound conditions.

### 13. List Comprehensions (Single-Variable, V1-V4)

`SESSION-076/List-Comprehension-General-Format-And-Some-Examples.py`

`SESSION-076/SINGLE-VARIABLE-LIST-COMPREHENSION-V1-TO-V4.pdf`

General syntax and four progressive versions of single-variable list comprehensions. From basic [expr for x in iterable] to filtered forms with conditions.

### 14. Lambda Expressions & map() Function

`SESSION-073/LAMBDA-IDLE-LOG.py`

`SESSION-073/power_lambda_map.py`

`SESSION-073/LAMBDA-IDLE-LOG-FINAL.pdf, MAP-IDLE-LOG.pdf`

Anonymous functions using lambda syntax. Using map() to apply functions across iterables. Combining lambda with map for concise data transformations.

### 15. filter() Function

`SESSION-074/IDLE-FILTER-LOG.pdf`

`SESSION-074/01-write-demo.py`

Using filter() with lambda and named functions to select elements from iterables based on conditions. Comparison with list comprehension filtering.

### 16. reduce() Function

`SESSION-073 to 077 / Functional programming sessions`

Reducing iterables to single values using functools.reduce(). Accumulation patterns and comparison with explicit loops.

### 17. Set Operations

`SESSION-059/SET-IDLE-LOG.py`

`SESSION-060/ Set continuation scripts`

Set data structure operations: union, intersection, difference, and symmetric difference. Set membership testing and mathematical set theory applied to Python.

### 18. File Handling & Text Processing

`SESSION-051/052 / 6 files each`

Advanced file handling techniques and text processing. Reading, writing, and manipulating text data from files.

## LEVEL 3 - INTERMEDIATE: OOP, Data Structures & File I/O

Object-oriented programming basics, complex data structures, file operations, and GUI development. Sessions 023-050.

### 19. OOP - Quadrilateral Class

`SESSION-090/03-Quadrilateral-Version-I.py`

`SESSION-091/01-Quadrilateral.py, 02-Quadrilateral.py`

Quadrilateral class with constructor accepting 4 sides and 4 angles. Includes validation, instance attributes, and progressive version improvements across sessions.

### 20. Class Definitions, Constructors & Methods

`SESSION-024/01-class-demo.py`

`SESSION-025/01-class-date-complete.py`

`SESSION-025 to 030 / OOP progression scripts`

Defining classes with the class statement, __init__ constructors, instance attributes, the self parameter, methods, and object instantiation. Progressive examples building a Date class.

### 21. Complex GUI Applications (Tkinter)

`GUI_USING_TKINTER/10_cpa_admission_form.py`

Complex multi-frame Tkinter application with 5 course selection radio buttons, checkbox groups, and a complete registration workflow with validation. Demonstrates real-world GUI application structure.

### 22. File I/O & Command-Line Arguments

`SESSION-050/01-create-and-write-on-file.py`

`SESSION-050/03-file-copy.py`

`SESSION-050/04-commandline-arguments.py`

Creating, writing, and copying files. Using sys.argv to accept command-line arguments. File modes and file object methods.

### 23. Data Structures: Lists, Tuples, Dictionaries

`SESSION-032 to 045 / Data structure scripts`

`SESSION-063/LIST-GENERAL-SYNTAX.py, LIST-IDLE-LOG.py`

List operations (indexing, slicing, append, extend, pop, insert), tuple immutability and packing/unpacking, dictionary key-value operations, and comparative use cases for each structure.

### 24. Global Variables & Scope Basics

`SESSION-023/global variable scripts`

Introduction to global variables, variable scope within functions, and the distinction between local and global namespaces.

### 25. GUI Widgets: Checkboxes, Comboboxes, Radio Buttons, Menus

`GUI_USING_TKINTER/07_check_box_demo.py`

`GUI_USING_TKINTER/08_combo_box_demo.py`

`GUI_USING_TKINTER/09_radio_button.py`

`GUI_USING_TKINTER/11_menu_demo.py`

Interactive widget types: Checkbutton for multi-select options, Combobox for dropdown selection, Radiobutton for exclusive choice groups, and Menu for application menu bars.

## LEVEL 2 - CORE: Functions, Control Flow & Branching

Fundamental programming constructs: function definitions, branching, looping, and basic algorithms. Sessions 010-021.

### 26. Algorithms - Insertion Sort

`SESSION-017/insertion_sort.py`

Implementation of the insertion sort algorithm. Demonstrates algorithmic thinking with nested loops, comparisons, and element shifting.

### 27. Boolean Operations & Comparisons

`SESSION-017/boolean scripts`

Boolean logic: and, or, not operators. Comparison operators (==, !=, <, >, <=, >=). Truthy and falsy values in Python.

### 28. Branching: if / elif / else

`SESSION-018/01-if-statement.py`

`SESSION-018/02-if-else-statement.py`

`SESSION-018/03-if-elif-else-statement.py`

Conditional execution with if statements, if-else for two-way branching, and if-elif-else for multi-way branching. Covers condition evaluation and code block indentation.

### 29. Looping: while, for, break, continue, pass

`SESSION-015, 016, 040-041 / Control flow scripts`

Iteration with while and for loops. Loop control statements: break to exit early, continue to skip iterations, pass as a placeholder. Iterating over ranges, lists, and other iterables.

### 30. Function Definitions (def statement)

`SESSION-010/01-def-statement-demo.py`

`SESSION-011/IDLE log files`

Defining reusable functions with def. Parameters and arguments, return values, function calls, and the basics of code organization through functions.

### 31. GUI Basics: Buttons, Events, Toggles

`SESSION-020/GUI button demos`

`SESSION-021/Toggle functionality scripts`

`GUI_USING_TKINTER/04_feet_to_meter.py`

`GUI_USING_TKINTER/05_compute_gravitational_gui.py`

Button widgets, event handling, toggle functionality, and practical calculator GUIs (unit conversion, gravitational force). Connecting UI elements to Python functions.

## LEVEL 1 - BEGINNER: Variables, Data Types, I/O & Setup

The foundation of Python programming. Installation, basic data types, variables, input/output, and the first GUI application. Sessions 006-009.

### 32. First GUI Application

`SESSION-009/01-first-gui-app.py`
`GUI_USING_TKINTER/01_widget_demo.py`
`GUI_USING_TKINTER/02_padx_pady_demo.py`
`GUI_USING_TKINTER/03_textvariable_demo.py`

Creating a first Tkinter window. Basic widgets: Label, Entry, Button. Layout management with padding (padx, pady). Variable binding with StringVar and IntVar.

### 33. Input/Output & Type Conversion

`SESSION-008/01-input-data.py`

Reading user input with input(). Displaying output with print(). Type conversion functions: int(), float(), str(). Getting data from the user and displaying results.

### 34. Arithmetic Operators & ASCII Codes

`SESSION_007/ARITHMETIC-OPERATORS-IDLE-LOG.py`
`SESSION_007/ASCII-CODES-IDLE-LOG.txt`

Basic arithmetic operators: +, -, *, /, //, %, **. Operator precedence. ASCII character codes and the ord()/chr() functions.

### 35. Data Types

`SESSION_006/BASICS.txt`
`WEEK_1_Material/5_DataTypes.pdf`

Python's built-in data types: bool, int, float, str, list, tuple, dict, set. Understanding type(), isinstance(), and dynamic typing.

### 36. Assignment Statements & Variables

`WEEK_1_Material/3_AssignmentStatementScenarios.pdf`
`WEEK_1_Material/4_AssignmentStatementAlgorithm.pdf`

How Python assignment works: binding names to objects. The object model, reference counting, and memory management. Assignment scenarios and the algorithm Python follows internally.

### 37. Python Installation & Setup

`WEEK_1_Material/1_Python_Installation_Guide.pdf`
`WEEK_1_Material/2_PythonFileStructure.pdf`
`IDLE_DOWNLOAD/python-3.13.6-amd64.exe`

Installing Python on Windows. Understanding Python file structure and organization. Setting up the IDLE development environment. The very first step of the learning journey.

## Quick Reference: Complete Sequence (Advanced to Simple)

### LEVEL 6

- 1. Metaclasses & Abstract Base Classes
- 2. Function Decorators (Production-Style)
- 3. Advanced Class Design - Vector3D
- 4. Advanced Class Design - Triangle
- 5. Code Object Verification & Attribute Management

### LEVEL 5

- 6. Closures & Function Factories
- 7. LEGB Scope Rule - Complete Mastery
- 8. global & nonlocal Statements
- 9. Advanced Parameter Passing (*args, **kwargs)
- 10. Nested Function Definitions
- 11. Exception Handling (try/except)

### LEVEL 4

- 12. Advanced List Comprehensions (Multi-Variable)
- 13. List Comprehensions (Single-Variable, V1-V4)
- 14. Lambda Expressions & map() Function
- 15. filter() Function
- 16. reduce() Function
- 17. Set Operations
- 18. File Handling & Text Processing

### LEVEL 3

- 19. OOP - Quadrilateral Class
- 20. Class Definitions, Constructors & Methods
- 21. Complex GUI Applications (Tkinter)
- 22. File I/O & Command-Line Arguments
- 23. Data Structures: Lists, Tuples, Dictionaries
- 24. Global Variables & Scope Basics
- 25. GUI Widgets: Checkboxes, Comboboxes, Radio Buttons, Menus

### LEVEL 2

- 26. Algorithms - Insertion Sort
- 27. Boolean Operations & Comparisons
- 28. Branching: if / elif / else
- 29. Looping: while, for, break, continue, pass
- 30. Function Definitions (def statement)
- 31. GUI Basics: Buttons, Events, Toggles

### LEVEL 1

- 32. First GUI Application
- 33. Input/Output & Type Conversion
- 34. Arithmetic Operators & ASCII Codes