I'll restructure the Python learning plan for 14 days with 1.5 hours daily, focusing on practical applications and micro-topics.

Day 1: Advanced Data Structures
Time Split:

- 20 min: Concept review
- 40 min: Coding practice
- 30 min: Mini-project

Micro Topics:

1. Lists & Tuples
   - List slicing techniques
   - Memory views
   - Tuple as dictionary keys
   - Packing/unpacking
   - List internals
2. Dictionaries
   - Dictionary views
   - Dictionary merging
   - defaultdict patterns
   - OrderedDict vs dict
   - Dictionary comprehension tricks

Practice Project: Build a caching system with LRU implementation

Day 2: Collections Module Deep Dive
Micro Topics:

1. Counter class
   - Most common elements
   - Element arithmetic
   - Counter updating
   - Subtraction patterns
2. deque operations
   - Rotation methods
   - Bounded vs unbounded
   - Thread-safe operations
   - Performance optimization

Practice Project: Build a log processor with rotating buffer

Day 3: Comprehensions & Generators
Micro Topics:

1. Comprehensions
    - Nested comprehensions
    - Set/dict comprehensions
    - Filter patterns
    - Error handling
    - Performance implications
2. Generator Expressions
    - Memory efficiency
    - Chaining generators
    - send() protocol
    - Closing generators
    - Exception handling

Practice Project: Create a memory-efficient data pipeline

Day 4: Decorators & Closures
Micro Topics:

1. Decorator Patterns
    - Function decorators
    - Class decorators
    - Method decorators
    - Parametrized decorators
    - Stacking decorators
2. Closure Implementation
    - Nonlocal variables
    - Cell objects
    - Binding behavior
    - Memory patterns

Practice Project: Build an authentication decorator system

Day 5: Advanced Functions
Micro Topics:

1. Function Objects

- Attributes
- Annotations
- Callable types
- Partial functions
- Function factories

2. Parameter Handling
   - Keyword-only args
   - Positional-only args
   - Variable arguments
   - Default mutable arguments
   - Type hints in functions

Practice Project: Create a command pattern implementation

Day 6: Context Managers
Micro Topics:

1. Context Protocol
   - **enter/exit**
   - Exception handling
   - Reentrant contexts
   - Async contexts
2. contextlib
   - @contextmanager
   - ExitStack
   - suppress
   - closing

Practice Project: Resource management system

Day 7: Advanced OOP - Part 1
Micro Topics:

1. Class Mechanics
   - Method resolution order
   - Super() patterns
   - Multiple inheritance
   - Abstract methods
   - Interface patterns
2. Special Methods

    - Comparison methods
    - Container methods
    - Attribute access
    - String representation

Practice Project: Build a custom container type

Day 8: Advanced OOP - Part 2
Micro Topics:

1. Descriptors
    - Property implementation
    - Lazy properties
    - Computed attributes
    - Validation descriptors
2. Metaclasses
    - Class creation
    - Attribute initialization
    - Class registration
    - Abstract base classes

Practice Project: Create an attribute validation system

Day 9: Error Handling Advanced
Micro Topics:

1. Exception Patterns
    - Custom exceptions
    - Exception chaining
    - Context managers
    - Cleanup patterns
2. Debugging
    - pdb usage
    - Logging patterns
    - Stack trace analysis
    - Debug decorators

Practice Project: Build an error reporting system

Day 10: Concurrency Basics
Micro Topics:

1. Threading
    - Thread lifecycle
    - Daemon threads
    - Thread pools
    - Race conditions
2. Synchronization
    - Locks
    - Events
    - Semaphores
    - Conditions

Practice Project: Multi-threaded download manager

Day 11: Advanced Concurrency
Micro Topics:

1. AsyncIO
    - Event loops
    - Coroutines
    - Task scheduling
    - AsyncIO patterns
2. Multiprocessing
    - Process pools
    - Shared memory
    - Pipes vs Queues
    - Process synchronization

Practice Project: Async web scraper

Day 12: File Operations
Micro Topics:

1. File Handling
    - Binary files
    - Memory mapping
    - File locking
    - Temporary files
2. Pathlib
    - Path operations
    - Glob patterns

    - File monitoring
    - Path protocols

Practice Project: File system monitor

Day 13: Data Processing
Micro Topics:

1. CSV/JSON/XML
    - Custom parsers
    - Schema validation
    - Streaming parsers
    - Data transformation
2. Serialization
    - Pickle protocol
    - Custom serializers
    - Security considerations
    - Performance optimization

Practice Project: Data format converter

Day 14: Testing & Documentation
Micro Topics:

1. Testing
    - Pytest fixtures
    - Mocking
    - Parametrization
    - Coverage analysis
2. Documentation
    - Docstring formats
    - Type hints
    - API documentation
    - Documentation testing

Practice Project: Create a test suite with documentation

Daily Practice Structure (1.5 hours):

1. 20 min: Quick theory and examples
2. 40 min: Hands-on coding exercises

3. 30 min: Project work