Here’s a for :

- Problem-solving concepts
- Problem definition & program design
- Debugging techniques & types of errors
- Documentation best practices

- Flowcharting & decision tables
- Writing algorithms
- Structured programming concepts
- Top-down vs. bottom-up methodologies

- Structure of a Python program
- Python interpreter & shell
- Using Python as a calculator
- Atoms, identifiers, keywords, literals, and strings
- Operators & expressions

- Branching: if, else, elif statements
- Looping: for & while loops
- Exit functions
- break, continue, and pass statements

- Understanding strings & string operations
- String indexing & slicing
- String functions & methods

- Introduction to lists & list indexing
- List operations: adding, removing, updating
- Built-in list functions & methods

- Understanding tuples & their properties
- Tuple operations & functions
- When to use tuples vs. lists

- Introduction to dictionaries
- Accessing, modifying, and deleting key-value pairs
- Dictionary functions & methods

- Defining & calling functions
- Types of functions (built-in & user-defined)
- Function arguments (default, keyword, variable-length)
- Anonymous (lambda) functions
- Global vs. local variables

- Creating & importing modules
- Using external libraries & packages
- Understanding and structuring Python projects

- Printing output on screen
- Reading input from the keyboard
- File handling: reading & writing files
- File handling functions

- Introduction to exceptions
- Handling errors with try-except
- Using finally blocks
- Creating user-defined exceptions
