DayTopicTasks1Introduction & SetupInstall Python, IDE (VS Code/PyCharm), Write First Program2Variables & Data TypesLearn int, float, str, list, tuple, dict, set3Operators & ExpressionsArithmetic, Logical, Comparison, Membership, Identity4Conditional Statementsif, elif, else5Loopsfor, while, break, continue6Mini-ProjectCalculator, Number Guessing Game	• We	Week 1: Python Basics		
2 Variables & Data Types Learn int, float, str, list, tuple, dict, set 3 Operators & Expressions Arithmetic, Logical, Comparison, Membership, Identity 4 Conditional Statements if, elif, else 5 Loops for, while, break, continue	Day	Торіс	Tasks	
3 Operators & Expressions Arithmetic, Logical, Comparison, Membership, Identity 4 Conditional Statements if, elif, else 5 Loops for, while, break, continue	1	Introduction & Setup	Install Python, IDE (VS Code/PyCharm), Write First Program	
4 Conditional Statements if, elif, else 5 Loops for, while, break, continue	2	Variables & Data Types	Learn int, float, str, list, tuple, dict, set	
5 Loops for, while, break, continue	3	Operators & Expressions	Arithmetic, Logical, Comparison, Membership, Identity	
	4	Conditional Statements	if, elif, else	
6 Mini-Project Calculator, Number Guessing Game	5	Loops	for, while, break, continue	
	6	Mini-Project	Calculator, Number Guessing Game	

Week 2: Functions & Data Structures Day Topic Functions Defining, Arguments, Return Values, Lambda Functions Factorial, Fibonacci Recursion Lists & Tuples CRUD Operations, Slicing, List Comprehension Dictionaries & Sets Keys, Values, Methods, Hashing String Manipulation Slicing, Methods, Regular Expressions 6 Mini-Project Contact Book, Password Generator

● Week 3: File Handling & OOPs				
Day	Торіс	Tasks		
1	File Handling	Read, Write, Append, With Statement		
2	Exception Handling	try, except, finally, raise		
3	Object-Oriented Programming	Class, Object, Methods		
4	OOPs Concepts	Inheritance, Polymorphism, Encapsulation, Abstraction		
5	Modules & Packages	Built-in & Custom Modules		
6	Mini-Project	To-Do List, Student Management System		

Week 4: Advanced Python & DSA Basics				
Day	Торіс	Tasks		
1	Iterators & Generators	yield, next()		
2	Decorators & Closures	@decorator, nested functions		
3	Data Structures Intro	Lists, Stacks, Queues		
4	Searching Algorithms	Linear Search, Binary Search		
5	Sorting Algorithms	Bubble, Selection, Insertion		
6	Mini-Project	Sorting Visualizer, Quiz App		

Week	Week 5: DSA (Intermediate)		
Day	Торіс	Tasks	
1	Linked Lists	Singly, Doubly, Circular	
2	Stacks	Push, Pop, Applications	
3	Queues	Simple, Circular, Deque	
4	Hashing	Hash Table, Collision Handling	
5	Trees	Binary Tree, BST	
6	Mini-Project	Expression Evaluator, File Organizer	

	● Week 6: Graphs & Final Project					
Торіс	Tasks					
Graphs	BFS, DFS					
Dynamic Programming	Fibonacci, Knapsack					
Advanced Algorithms	Dijkstra, Floyd Warshall					
Final Revision	Solve Random Problems					
Capstone Project	Implement Final Project					
Final Project Submission	Explanation & Documentation					
	Graphs Dynamic Programming Advanced Algorithms Final Revision Capstone Project					

Resources to Learn From Python Basics & Advanced: Python Docs, W3Schools, Real Python Data Structures & Algorithms (DSA): GeeksforGeeks, LeetCode, CodeChef Practice Problems: HackerRank, LeetCode, CodeWars Projects: Kaggle, PythonProjects

Weekly Mini-Projects

- Week 1: Calculator, Number Guessing Game
- Week 2: Contact Book, Password Generator
- Week 3: To-Do List, Student Management System
- Week 4: Sorting Visualizer, Quiz App
- Week 5: Expression Evaluator, File Organizer
- Week 6: Final Project (Choose a real-world problem)

Learning Approach

- 1. Concept + Code + Explain
 - Learn the concept
 - Write and practice code
 - Try explaining to yourself or a friend
- 2. Solve Problems Daily
 - At least 5-10 questions per concept
- 3. Project-Based Learning
 - Implement mini-projects to reinforce learning