

Course Overview:

This intensive 60-hour course is designed to equip learners with the skills necessary to secure a role as a **Python Developer** in leading IT companies. The curriculum blends **core programming concepts**, **data structures**, **project development**, and **interview preparation**, ensuring job readiness through hands-on experience and real-world coding challenges.

Course Objectives:

By the end of this course, learners will be able to:

- Write clean, efficient Python code for real-world applications.
- Solve algorithmic challenges commonly asked in coding interviews.
- Develop full-stack web applications using Flask and SQLite.
- Use Git and GitHub for version control and collaborative development.
- Confidently clear technical rounds of IT placement processes.

Target Audience:

- Final year college students or recent graduates seeking IT placements.
- Freshers looking to upskill in Python for software developer roles.
- Anyone with basic programming knowledge aiming for a Python career.

Course Duration:

- **Total Hours:** 60 hours
- **Schedule Options:**
 - 2 hours/day for 30 days (weekday batch)
 - 3 hours/day for 20 days (intensive batch)
 - Weekend batch: 5 hours/week for 12 weeks

Course Structure:

Module 1: Python Programming Fundamentals

Objective: Build a strong foundation in Python syntax, data structures, and control flow.

Topics:

- Python Installation and Setup (with VSCode or PyCharm)
- Variables, Data Types, Type Casting
- Operators (Arithmetic, Logical, Comparison)
- Conditional Statements (if, else, elif)
- Loops (for, while, break, continue)
- Functions (arguments, return values, scope, recursion)
- Built-in Functions & Modules
- Exception Handling (try, except, finally, raise)

Practice:

- Simple calculator
- Prime number generator
- Palindrome checker

Module 2: Data Structures & Algorithms in Python

Objective: Master Python's built-in data structures and common algorithms.

Topics:

- Lists, Tuples, Sets, Dictionaries

- List Comprehensions
- String Manipulation
- Sorting and Searching Algorithms
- Time and Space Complexity
- Recursion and Iterative Approaches
- Stacks, Queues, Linked Lists (using lists or custom classes)
- Trees and Graphs (basic intro + implementation)
- Hashing and Hash Tables

Practice:

- LeetCode, HackerRank, Codeforces challenges
- Implement DSA concepts using Python

Module 3: Object-Oriented Programming (OOP) & File Handling

Objective: Learn clean, modular, and reusable code structure using OOP.

Topics:

- Classes and Objects
- `__init__`, `__str__`, `__repr__`
- Encapsulation, Inheritance, Polymorphism, Abstraction
- Class and Static Methods
- File Handling (text, CSV, JSON)
- Exception Handling with OOP

Practice:

- Inventory management system

- Library/book management app

Module 4: Python for Real-World Development

Objective: Apply Python in development environments, understand web and automation.

Topics:

- Virtual Environments & pip
- Modules & Packages
- Introduction to Flask or Django
- REST APIs in Flask/Django (GET, POST, PUT, DELETE)
- SQLite / PostgreSQL with SQLAlchemy or Django ORM
- Web scraping with BeautifulSoup or Selenium
- Automation with os, shutil, schedule, requests

Practice:

- CRUD app with Flask/Django
- Web scraper for job listings or weather
- Automate email sending or file cleanup

Module 5: Testing, Debugging & Version Control

Objective: Adopt industry practices for testing and collaboration.

Topics:

- Debugging techniques

- Unit Testing with unittest / pytest
- Writing test cases
- Git and GitHub (clone, commit, push, pull, branches)
- Writing README and documentation

Module 6: Projects

Objective: Build portfolio projects and practice coding interviews.

Projects:

- E-commerce backend with Flask/Django
- Chatbot using Python (with OpenAI or basic rules)
- File organizer / Personal assistant script