SMART TRAINERS

LESSON PLAN FOR
PYTHON PROGRAMMING COURSE

COURSE TITLE:	**Basic to Intermediate Python Programming**	CREDIT HOURS:	**02**
COURSE CODE:	**PY 003**	DURATION:	**08 Weeks (03 Classes Each Week)**

COURSE INSTRUCTOR: Aelia shah	**Batch: 04**	**Course Starting Date:**	**Course Suspension Date:**

COURSE LEARNING OUTCOMES: Upon successful completion of this course, the student will be able to: Set up a Python development environment and write, run, and debug Python programs confidently. Apply core language constructs — variables, operators, control flow, functions, and modules — to solve everyday programming problems. Work with Python's built-in data structures (lists, tuples, sets, dictionaries) and perform file I/O and structured exception handling. Apply object-oriented programming (OOP) principles — classes, inheritance, encapsulation, and polymorphism — to design clean, reusable software. Leverage advanced Python features such as decorators, generators, comprehensions, and the standard library. Use third-party libraries (NumPy, Pandas, Requests) for data processing, automation, and API integration. Build practical CLI applications and web-scraping scripts, and understand pathways into Data Science, Backend Development, or Automation.

CLO	DESCRIPTION	Learnings
1. **Python Fundamentals** 2. **Control Flow & Functions** 3. **Data Structures** 4. **File I/O & Exceptions** 5. **Modules & Packages**	Understand Python syntax and environment setup. Write programs using variables, operators, and control structures. Build reusable functions and work with Python's built-in data structures. Handle files and errors gracefully using exception handling. Organise code into modules and packages.	1. Python Setup & Syntax 2. Variables, Operators & Data Types 3. Control Flow (if/else, loops) 4. Functions & Lambda 5. Lists, Tuples, Sets, Dicts 6. File I/O & Exception Handling
6. **OOP in Python** 7. **Advanced Python Features** 8. **Standard Library & Third-Party Packages** 9. **Data Handling with Pandas & NumPy**	Apply object-oriented design principles using Python classes. Explore advanced language features such as decorators, generators, and comprehensions. Leverage the standard library and install third-party packages. Process and analyse data with NumPy and Pandas.	1. Classes, Objects & Inheritance 2. Polymorphism & Encapsulation 3. Decorators & Generators 4. List/Dict Comprehensions 5. os, sys, datetime, json Modules 6. NumPy Arrays & Pandas DataFrames

Contents	Week	CLO / Week	Marks %	Delivery Methods	Assessment Methods
PHASE 1 — BASIC PYTHON (Weeks 1 – 4)					
1. **Python Environment Setup** · Installing Python & VS Code / PyCharm · Python REPL and running scripts · Understanding PIP and virtual environments · First Python program: Hello, World! 2. **Variables, Data Types & Operators** · Numeric, string, boolean & None types · Type conversion and casting · Arithmetic, comparison, logical operators · Input / Output basics (input(), print())	Week 1	1–2	5%	Online Lectures & Practical Work	Practical Project
3. **Strings in Depth** · String methods and formatting · f-strings and template strings · Slicing, indexing, and string immutability 4. **Control Flow** · if, elif, else — nested conditions · for loops, range(), while loops · break, continue, pass statements	Week 2	3	5%	Online Lectures & Practical Work	Practical Project
5. **Functions** · Defining and calling functions · Parameters, arguments, default values · Return values and scope (local vs global) · Lambda (anonymous) functions & recursion basics 6. **Lists & Tuples** · Creating, indexing, and slicing lists · List methods: append, pop, sort, reverse · List comprehensions · Tuples, immutability, unpacking & packing	Week 3	4	10%	Online Lectures & Practical Work	Practical Project

7. Dictionaries & Sets • Creating and accessing dicts • Dict methods: keys, values, items • Dict comprehensions & nested structures • Sets and set operations **8. File I/O & Exception Handling** • Opening, reading, and writing files • Context managers (with statement) • Working with CSV and JSON files • try / except / finally, raising custom exceptions	Week 4	5	10%	Online Lectures & Practical Work	Practical Project
PHASE 2 — INTERMEDIATE PYTHON (Weeks 5 – 8)					
9. Object-Oriented Programming — Basics • Classes and objects; constructors (init) • Instance vs class attributes • Methods and self keyword • Encapsulation & properties (@property) **10. OOP — Advanced** • Inheritance and super() • Method overriding & polymorphism • Dunder / magic methods (str, repr, len) • Abstract classes (ABC)	Week 5	6	10%	Online Lectures & Practical Work	Practical Project
11. Advanced Python Features • Iterators & generators (yield) • Decorators and closures • List / dict / set comprehensions (advanced) • *args, **kwargs, and context managers **12. Modules, Packages & Standard Library** • Importing modules & creating packages • os, sys, math, datetime modules • json, csv, re (regex) modules • Virtual environments & PIP deep-dive	Week 6	7	10%	Online Lectures & Practical Work	Practical Project

13. NumPy & Pandas Basics • NumPy arrays, operations & slicing • Pandas Series and DataFrames • Reading CSV/Excel files with Pandas • Data cleaning, filtering & descriptive stats **14. Web & API Interaction** • HTTP basics (GET, POST) & requests library • Consuming public REST APIs • Parsing JSON responses • Introduction to web-scraping with Beautiful Soup	Week 7	8	20%	Online Lectures & Practical Work	Practical Project
15. Capstone Projects • CLI To-Do application (Basic concepts) • OOP-based Library Management System • Data analysis mini-project with Pandas • Weather / News API integration app **16. Course Review & Next Steps** • Code review & best practices walkthrough • Project presentations and Q&A; • Pathways: Data Science, Backend Dev, Automation • Recommended resources & roadmap	Week 8	8–9	30%	Online Lectures & Practical Work	Practical Project

The End