

Day (Classes)	TOPICS	PARTICULARS	LEARNING OBJECTIVES	LEVEL
1 - 2	Python Basics	Aspects of language (primitive constructs, syntax, semantics), understanding a program in general, standard commands, writing the first Python program, and printing text/output.	Student will get an overview of Python and be able to write basic commands.	Basic
1 - 2	Simple Operations	Calculations, handling "dividing by zero," data types (int, float, bool, NoneType), type() function, type casting, exponentiation, quotients, remainders, and Programming vs. Math.	Student will perform math operations and understand Python's primitive constructs.	Basic
2 - 3	String Operations	String manipulation, backslashes, newlines, concatenation, variables, naming conventions, str(), int(), float(), input(), len(), indexing, and slicing.	Student will be able to work with Python variables and process text input.	Basic
3 - 4	PROJECT 1: Logic Games	Build: A multi-function calculator that handles complex arithmetic and error trapping (e.g., division by zero).	Student will apply basic syntax to a functional software tool.	Basic (Build)
2 - 3	Control Structures	Booleans, comparison and assignment operators, if/else/elif statements, proper indentation, nested conditionals, boolean logic/operators, compound booleans, and operator precedence.	Student will master boolean logic and decision-making operations.	Basic
3 - 4	Looping Structures	Automating tasks with while loops, for loops, the range() function, and control statements including break and continue.	Student will be able to automate tasks using efficient looping structures.	Basic
3 - 4	PROJECT 2: Adventure Engine	Build: A text-based "Choose Your Own Adventure" game using nested conditionals and looping structures for game flow.	Student will manage complex program logic through user input.	Basic (Build)
3 - 4	Functions & Modules	Code reuse, invoking functions, function characteristics (name, parameters, arguments, body, docstring), comments, functions as objects, the standard library, and pip.	Student will understand functional programming and modular code reuse.	Intermediate
3 - 4	PROJECT 3: Smart Text Analyzer	Build: An automated tool that calculates word frequency, character count, and sentence length using functions and strings.	Student will use modular code to process and analyze data.	Intermediate (Build)
4 - 5	Core Data Structures	Tuples (mutability, slicing), Lists (operations, mutation, iteration, nesting, aliasing, and cloning), and Dictionaries (operations and functions).	Student will manage and handle complex collections of data effectively.	Intermediate
3 - 4	Exceptions & Files	Error handling with try/except/finally, raising exceptions, assertions, opening files, reading content, writing data, and persistent data storage.	Student will handle program errors and manage external file storage.	Intermediate

3 - 4	PROJECT 4: Record Manager	Build: A student record system that uses dictionaries and File I/O to save and load student data between sessions.	Student will build persistent data tools that survive program restarts.	Intermediate (Build)
4 - 5	Object Oriented Programming (OOP)	Classes, data abstraction, inheritance, magic methods, operator overloading, data hiding, and interfaces.	Student will master the OOP paradigm used in professional software engineering.	Advanced
Row	SEPARATOR	ADVANCED & PROFESSIONAL LEVEL	DATA SCIENCE & MACHINE LEARNING	---
5 - 6	Numpy & Pandas	Numpy: Indexing/slicing arrays, assigning values, and vectorized operations. Pandas: DataFrames, indexing, summary statistics, and Groupby/Aggregation.	Student will clean and manipulate real-world datasets for high-performance math.	Advanced
4 - 5	PROJECT 5: Global Stats Visualizer	Build: Analyze a public dataset (e.g., climate data) and create professional graphs using Matplotlib and Pandas .	Student will perform visual data storytelling for technical reporting.	Advanced (Build)
6 - 8	Machine Learning: Regression	Exploratory Data Analysis (EDA), correlation, fitting univariate and multivariate linear regressions, and model evaluation.	Student will build predictive models based on historical mathematical trends.	Professional
6 - 8	PROJECT 6: Trend Predictor	Build: A machine learning model that predicts future outcomes (e.g., sales or weather) based on historical data files.	Student will implement their first functional machine learning model.	Professional (Build)
6 - 8	ML: Classification	Data visualization for ML, classification modelling, prediction, and model evaluation with new test data.	Student will build AI that can categorize data into distinct labels.	Professional
6 - 8	PROJECT 7: Pattern Recognition Tool	Build: A classification application that identifies patterns (e.g., detecting spam or identifying object types).	Student will master advanced ML modelling and performance evaluation.	Professional (Build)
8 - 12	ML: Clustering & Capstone	Centroid-based clustering (e.g., wine data), label prediction, and the Final Professional Capstone Build .	Student will design, build, and Showcase a professional-grade AI solution.	Professional (Capstone)