Python

Unit-wise syllabus and marks for the 2026-27 session.

UNITS	SESSION
12	2026-27

Exam pattern. Board-independent programming track, so there is no fixed external exam. Assessment is project- and practical-driven: roughly 60% hands-on coding assignments and projects, 40% concept checks/quizzes. For reference, in CBSE Computer Science (which uses Python) the board pattern is 70 marks theory + 30 marks practical.

Practical. Hands-on lab in every unit: learners code along in IDLE/VS Code and complete graded programming assignments. Culminates in multiple projects, console apps (calculator, quiz, game), a file-processing automation script, an OOP-based application, and a small data analysis project on a real CSV dataset using Pandas/Matplotlib. Mirrors CBSE-style practical work (program writing + project + viva).

#	UNIT AND TOPICS	MARKS
1	**Getting Started with Python** Installing Python and using the interpreter / IDE (IDLE, VS Code); Interactive mode vs script mode; print() and basic output; Comments and code readability; Your first program	–
2	**Variables, Data Types and Operators** Variables and assignment; Numbers (int, float, complex) and Booleans; Type casting and type checking; Arithmetic, comparison, logical, assignment operators; Identity and membership operators; Taking user input with input()	–
3	**Strings** Creating and indexing strings; Slicing and immutability; String concatenation and repetition; Common string methods; String formatting (f-strings, format()); Escape sequences	–
4	**Control Flow** if / elif / else statements; Nested and shorthand conditionals; match-case statements; while loops; for loops and the range() function; break, continue and pass	–
5	**Data Structures** Lists: indexing, slicing, methods, sorting; List comprehensions; Tuples and unpacking; Sets and set operations; Dictionaries and nested dictionaries; Looping techniques over collections	–

#	UNIT AND TOPICS	MARKS
6	**Functions** Defining and calling functions; Parameters, arguments and return values; Default, keyword, *args and **kwargs; Scope: local vs global; Lambda (anonymous) functions; Recursion and an idea of efficiency; Modular thinking and reusability	–
7	**Modules and Standard Library** Importing modules and packages; math, random, statistics modules; datetime and time; Creating your own modules; pip and installing third-party packages; Brief tour of the standard library	–
8	**Input/Output and File Handling** Reading and writing text files; File modes (r, w, a, r+); with statement and context managers; Working with CSV files; Introduction to JSON	–
9	**Errors and Exception Handling** Syntax errors vs runtime errors; try / except / else / finally; Catching specific exceptions; Raising exceptions; Debugging strategies	–
10	**Object-Oriented Programming** Classes and objects; The __init__ method and self; Instance attributes and methods; Encapsulation and private members; Inheritance and method overriding; Polymorphism; Special / dunder methods (__str__, __repr__)	–
11	**Working with Data (Intro Libraries)** Introduction to NumPy arrays; Introduction to Pandas DataFrames; Basic data analysis on CSV data; Simple plotting with Matplotlib; When and why to use libraries	–
12	**Projects and Applications** Console mini-projects (calculator, quiz, number game); Text/file processing automation script; OOP-based application (e.g. bank/library/inventory manager); A small data-driven project using a CSV dataset; Code organisation, testing and documentation; Version basics and sharing your code	–

Need help covering this syllabus?

Live online classes with Kajal Mehta, teaching since 2004. Free demo first, no payment. WhatsApp +91 97255 25791 · kwickprep.com