
SYLLABUS FOR BOARD OF STUDIES ADOPTION — generated from the live catalogue record

Python Programming Foundations

C-02 · 3 days (configurable 1–5 days) · Informatics · Software Engineering · Years: 1st, 2nd

CONTENTS

1. Overview	2
2. Hours and credit position	2
3. Course outcomes	2
4. Course content — modules	2
5. Assessment scheme (100 marks)	2
6. CO–PO mapping	2
7. Every participant receives	3
8. Fee	3
9. Designed by	3

Catalogue C-02 · version 1 · generated 2026-08-24

OVERVIEW

From zero to writing and debugging real programs — data types, control flow, functions, files, libraries.

The shared foundation under all ten of your Computer Science specialisations, and the right first workshop for a first-year cohort.

HOURS AND CREDIT POSITION

Contact hours: 24 Guided project: 6 Self-study: 0 Notional total: 30

Credit claim: 1

30 notional hours ÷ 30 hours per credit (NSQF track, UGC Micro-credentials Guidelines §5.3.3) = 1 credits.

COURSE OUTCOMES

CO1. Write programs that read input, branch on conditions and loop over data (K3)

CO2. Apply lists and dictionaries to organise and query real records (K3)

CO3. Decompose a problem into functions with clear parameters and return values (K4)

CO4. Process text and CSV files using the standard library (K3)

CO5. Debug a failing program by reading its traceback and isolating the fault (K4)

COURSE CONTENT — MODULES

Hours shown are the recommended format; module time scales proportionally when the engagement runs 1–5 days.

Module 1 — First programs (4h recommended)

Theory: Interpreter, variables, data types, input/output, errors as information

Lab: Twenty micro-exercises from print to formatted output

Output: Exercise notebook

Module 2 — Making decisions (4h recommended)

Theory: Booleans, comparisons, if/elif/else, while and for loops

Lab: Number games and pattern printers

Output: Four working control-flow programs

Module 3 — Collections (4h recommended)

Theory: Lists, tuples, dictionaries, sets; slicing; iteration patterns

Lab: Build a marks-register that answers class queries

Output: Marks-register program

Module 4 — Functions and structure (4h recommended)

Theory: Parameters, return values, scope, docstrings, decomposing a problem

Lab: Refactor day-one code into clean functions

Output: Refactored module

Module 5 — Files and libraries (4h recommended)

Theory: Reading and writing files, CSV, the standard library, installing packages

Lab: Process a real CSV dataset end to end

Output: File-processing utility

Module 6 — Debugging and capstone (4h recommended)

Theory: Tracebacks, print-debugging, common error patterns; capstone build

Lab: Capstone: a small utility each student designs and defends

Output: Capstone program + walkthrough

ASSESSMENT SCHEME (100 MARKS)

Continuous lab exercises — 40 marks

Capstone project & demonstration — 40 marks

Quiz & viva voce — 20 marks

CO-PO MAPPING

CO1 !' PO1 (strength 2)

Control flow is the foundation of computational thinking.

CO2 !' PO2 (strength 2)

Choosing the right collection is problem analysis in miniature.

CO3 !' PO3 (strength 2)

Decomposition into functions is design, not typing.

CO4 !' PO5 (strength 2)

File and library work is how Python meets real data.

CO5 !' PO12 (strength 2)

Reading errors independently is what makes a self-sufficient programmer.

EVERY PARTICIPANT RECEIVES

- Certificate of completion
- Course material
- LMS access
- Interview question bank
- Mock interview & viva practice
- Optional nasscom NSQF assessment (priced per candidate on your quotation)

FEE

11,800 per student, incl. GST

Excludes the optional nasscom assessment where offered. Final pricing on your college's quotation.

DESIGNED BY

Venkat Kurela — Founder & Director, Kurela Cognisive Private Limited