

EC-110: C AND PYTHON PROGRAMMING LAB

Course Title	C and Python Programming Lab	Course Code	EC-110
Year	I	Course Group	Practical
Teaching Scheme in Hrs (L: T:P)	1:0:2	Credits	2.0
Methodology	Lecture + Practical	Total Contact Hours	90
CIE	60 Marks	SEE	40 Marks

Prerequisites: This course requires the Basic knowledge Of Mathematics, Logical, Reasoning and IT skills

Course Outcomes: Upon completion of the course the student shall be able to

CO	Course Outcome
CO1:	Use basic computer applications and programming tools to perform simple computing tasks.
CO2:	Develop simple programs in C language using fundamental programming concepts.
CO3:	Write basic Python programs to solve simple problems using logical thinking.

S.NO	UNIT	PERIODS
1	Basic Computer & Programming Tools	18
2	C Programming Fundamentals	36
3	Python Programming Fundamentals	36
Total		90

Recommended books

Computer Fundamentals

1. P.K. Sinha & Priti Sinha – *Computer Fundamentals*

C Programming

1. E.Balagurusamy–Programming in ANSI C
2. Byron Gottfried – Programming with C

Python Programming

1. Reema Thareja – Python Programming Using Problem Solving Approach
2. Charles Severance – Python for Everybody

Suggested e-learning resources

1. <https://www.tutorialspoint.com/cprogramming>
2. <https://docs.python.org/3/tutorial/>
3. <https://www.w3schools.com>
4. <https://www.programiz.com>
5. <https://www.onlinegdb.com>
6. <https://www.programiz.com/python-programming/online-compiler>

Suggested Learning Outcomes:

Upon completion of the course, the student shall be able to

CO1: Use basic computer applications and programming tools to perform simple computing tasks.

1. MS-Word: Creating, saving, and formatting a document
2. MS-Word: Spell check, page setup, and printing
3. MS-PowerPoint: Creating slides and inserting text and images
4. MS-PowerPoint: Slide transitions, animations, and slide show
5. MS-Excel: Creating worksheets, simple formulas, charts, and printing
6. Familiarization with programming tools (GCC Compiler / Python IDLE)

CO2: Develop simple programs in C language using fundamental programming concepts.

1. Exercise on Structure of a C program and compilation process
2. Exercise on Keywords, identifiers, constants, and variables
3. Exercise on Data types and operators
4. Exercise on Input and output statements
5. Exercise on Formatted input and output
6. Exercise on Conditional statements: if, if-else
7. Exercise on else-if ladder and switch case
8. Exercise on Looping statements: while, for, do-while
9. Exercise on One-dimensional arrays
10. Exercise on Two-dimensional arrays
11. Exercise on String handling basics
12. Exercise on User-defined functions

CO3: Write basic Python programs to solve simple problems using logical thinking.

1. Exercise on Input and output statements
2. Exercise on Conditional statements
3. Exercise on Looping statements
4. Exercise on Functions in Python
5. Exercise on recursive Functions in Python (factorial, Fibonacci)
6. Exercise on Lists and tuples
7. Exercise on Dictionary basics
8. Exercise on Exceptions handling
9. Write a program to define class with its members and create instances of class
10. Exercise on Basic file handling (Open, read and write text files)
11. Demonstrate the installation of various packages and modules numpy,pandas,and matplotlib
12. Write a program using the numpy package in python.
13. Write a program using the pandas package in python.
14. Write a program using matplotlib package in python.
15. Write a program to validate data using regular expressions.

Suggested student Activities:

Student activity like mini-project, surveys, quizzes, etc. should be done in groups of 5-10 students.

1. Each group should do any one of the following types of activity or any other similar activity related to the course with prior approval from the course coordinator and programme coordinator concerned.
2. Each group should conduct different activities and no repetition should occur.
3. Explore and analyse topics to improve the level of creativity and analytical skill by taking Quiz/ tests/ assignments. Documents have to be maintained as a record.
4. Create a power point presentation on the topic relevant to the course or advanced topic as an extension to the course to improve the communication skills. Documents have to be maintained as a record.
5. Visit different sites relevant to topics. Listen to the lectures and submit a handwritten report
6. Attend Coding competitions

CO-PO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO 7
CO 1	2	1	1	3	1	0	1
CO 2	3	3	2	2	0	0	1
CO 3	3	3	2	2	0	0	1

CO–PO Justifications

CO1–PO1:

Students apply basic computer knowledge and fundamentals.
This strengthens foundational engineering understanding.

CO1–PO4:

Use of operating systems and programming tools is emphasized.
Students perform simple tests and executions.

CO2–PO1:

C programming uses logical and mathematical foundations.
Core engineering problem-solving skills are developed.

CO2–PO2:

Students analyze simple problems and convert them into programs.
Standard programming constructs are applied.

CO2–PO3:

Programs are designed to meet given requirements.
Structured solution development is practiced.

CO-PO Map

CO3–PO2:

Python programs enhance logical thinking and analysis.
Students solve well-defined problems effectively.

CO3–PO3:

Simple problem solutions are designed using Python.
Algorithmic thinking is developed.

CO3–PO7:

Python encourages learning modern and evolving technologies.
Students develop self-learning ability.