GANPAT UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

Programme	Bachelor of Technology	Branch/Spec.	All
Semester	II	Version	1.0.0.0
Effective from Academic Year	2026-27	Effective from the batch admitted in	July 2026
Course Code	2ESC1110	Course Name	Python Programming
Course Category	Engineering Science Courses (ESC)		

Teaching Scheme						Examination scheme (Marks)			
(Per week)	Lecture (DT)		Practical (Lab.)		Total		CE	SEE	Total
	L	TU	P	TW					
Credit	3	0	1	0	4	Theory	50	50	100
Hours	3	0	2	0	5	Practical	25	25	50

Pre-requisites:

Basic knowledge of programming concepts and problem-solving skills.

Course Outcomes

COs	Description
CO1	Understand Python fundamentals including data types, operators, control structures, and structured data types for problem-solving.
CO2	Apply functional programming concepts including functions, recursion, lambda expressions, and modules to develop modular programs.
CO3	Implement file handling operations for reading, writing, and processing data from various file formats.
CO4	Design and develop object-oriented programs using classes, inheritance, polymorphism, and exception handling mechanisms.

Theory Syllabus

Unit	Content	Hours
1	**Python Fundamentals** Why Python? Comparison with C (no semicolons, no braces, indentation matters), Installing Python and IDE (VS Code, PyCharm), Variables and data types (int, float, str, bool), Type conversion, Input/Output (input(), print()), Operators and expressions.	05
2	**AI Tools Introduction** What are AI coding assistants? (ChatGPT, Claude, GitHub Copilot), How do they work? (simple explanation of LLMs), Setting up AI tools in VS Code, Your first AI-assisted program, When to use AI and when NOT to use (building understanding first).	05
3	**Control Structures & Functions** if-elif-else (comparison with C's if-else), for loop and range(), while loop, break and continue, List comprehensions, Defining functions (def), Parameters and return values, Default parameters, *args and **kwargs (introduction), Lambda functions (simple use cases). Using AI to generate functions, Understanding and modifying AI-generated code, Asking AI to explain code.	06
4	**Python Data Structures** Creating and accessing lists (comparison with C arrays), List methods (append, insert, remove, sort), Slicing, Tuples (immutable lists), When to use list vs tuple, Creating dictionaries (key-value pairs), Dictionary methods, Sets (unique	07

	elements), Set operations (union, intersection), String methods (much easier than C!), String formatting (f-strings), Regular expressions introduction. Using AI for data manipulation tasks, Generating test data with AI	
5	**Object-Oriented Programming & File Handling** Classes and Objects (real-world analogy: Car class, specific cars as objects), __init__ method (constructor), Instance variables and methods, Inheritance (basic), Encapsulation (public/private - simple introduction), Opening files (with statement), Reading and writing text files, Working with CSV files, JSON basics. Using AI to design classes, AI-assisted file processing scripts	07
6	**Python Libraries & AI-Powered Development** NumPy basics (arrays, operations), Pandas basics (DataFrames, reading CSV), Matplotlib basics (simple plots), Installing libraries with pip	05
7	**Effective Prompt Engineering** Writing clear prompts for code generation, Iterative refinement (improving AI responses), Context management (what to tell the AI), Multi-step problem decomposition	05
8	**Critical Evaluation & Ethics** Identifying AI errors and hallucinations, Testing AI-generated code, When AI is wrong (edge cases), Academic integrity guidelines, Copyright considerations, building skills while using AI assistance	05

Practical and Self Learning Content
Practical, assignments, quiz, industrial visit, field survey and tutorials are based on the above syllabus.

Text Books	
1	“Learning program to Python” by Cody Jackson.
2	“Python Programming: An Introduction to Computer Science” by John Zelle, Franklin, Beedle & Associates

Reference Books	
1	“Introduction to Computation and Programming Using Python” by John V Guttag, Prentice Hall of India.
2	“Beginning Python” by Magnus Lie Hetland, Apress Publication.
3	“Beginning Python” by James Payne, Wrox Publication.
4	"Learn AI-Assisted Python Programming" by Leo Porter & Daniel Zingaro, Manning

ICT/MOOCs Reference	
1	https://nptel.ac.in/courses/106106145/
2	https://nptel.ac.in/courses/106106182/
3	Prompt Engineering for ChatGPT - Vanderbilt University (Coursera)
4	CS50P: Introduction to Programming with Python (Harvard, Free on YouTube)

Mapping of COs, POs, and PSOs														
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	0	2	0	0	1	0	1	0	2	1	1
CO2	3	2	2	1	2	0	0	1	1	1	0	2	1	1
CO3	2	1	2	1	3	0	1	1	0	1	0	2	2	1
CO4	3	3	3	1	2	0	2	2	1	2	1	3	2	2

Bloom's Taxonomy Level				
Unit	**Unit Title**	**Aligned COs**	**Learning Hours**	**BTL Level**
1	Python Fundamentals	CO1	05	R,U
2	AI Tools Introduction	CO1	05	U
3	Control Structures & Functions	CO1, CO2	06	A
4	Python Data Structures	CO1, CO2	07	A
5	Object-Oriented Programming & File Handling	CO3, CO4	07	A,N
6	Python Libraries & AI-Powered Development	CO2, CO3	05	A
7	Effective Prompt Engineering	CO2, CO4	05	A,C
8	Critical Evaluation & Ethics	CO4	05	E

Note:

- Version 1.0.0.0 (First Digit= New syllabus/Revision in Full Syllabus, Second Digit=Revision in Teaching Scheme, Third Digit=Revision in Exam Scheme, Forth Digit= Content Revision)
- 1 Hour Lecture = 1 Credit, 1 Hour Tutorial = 1 Credit, 2 Hours Practical = 1 Credit, 2 Hours Internship/Project/Seminar = 1 Credit
- Bloom's Taxonomy Level (BTL): R: Remember, U: Understand, A: Apply, N: Analyze, E: Evaluate, and C: Create