

<b>GANPAT UNIVERSITY</b>			
<b>FACULTY OF DIPLOMA ENGINEERING</b>			
Programme	Diploma in Computer Engineering/ Information Technology		
Semester	IV	Version	1.0.0.0
Effective from Academic Year	2025-26	Effective for the batch Admitted in	JULY 2025
Course Code	ICEIT4106	Course Name	Advanced Python programming

<b>I. TEACHING-LEARNING AND ASSESSMENT SCHEME</b>																			
Course Type	Course code	Course Title	Teaching & Learning Scheme								Examination Scheme								
			Credit				Actual Contact Hrs/week			SLH	Total Learning Hrs/Week	TH			PR			SLA	Total
			CL	TL	LL	Total	CL	TL	LL			CE	SEE-	Total	CE	SEE	Total		
DSC	ICEIT4106	Advanced Python programming	-	-	1	1	-	-	2	-	2	-	-	-	30	20	50	-	50

<b>Abbreviation:</b>	<b>CL-</b> Classroom Learning	<b>TL-</b> Tutorial Learning	<b>LL-</b> Laboratory Learning
	<b>SLH-</b> Self Learning Hours	<b>SLA -</b> Self Learning Assessment	<b>CE -</b> Continuous Evaluation
	<b>SEE –</b> Semester End Examination		

## **II. PRE-REQUISITES**

Completion of Programming in Python (basics including data types, control structures, functions, OOP, file handling, exception handling).

## **III. INDUSTRY /EMPLOYER EXPECTED OUTCOMES**

Develop practical Python applications for automation, data processing, simple web interfaces, and database handling - skills required for junior developer, automation engineer, and data entry/processing roles.

## **IV. COURSE LEARNING OUTCOMES**

At the end of the course, students will be able to:

- CO1.** Use advanced built-in features and third-party libraries for efficient programming.
- CO2.** Perform robust file handling and error management in applications.
- CO3.** Create simple graphical user interfaces and visualizations.
- CO4.** Process and analyse structured data using popular libraries.
- CO5.** Build basic web applications and connect to databases.

## **V. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL**

SR. NO	PRACTICAL/LABORATORY LEARNING OUTCOME(LLO)	PRACTICAL TITLES	RELEVANT COs
1	<b>LLO 1.1</b> Use advanced built-in modules from the collections library.	Implement programs using Counter, defaultdict, namedtuple.	CO1
2	<b>LLO 2.1</b> Apply functional programming tools from itertools and functools.	Programs using permutations, combinations, map, filter, reduce.	CO1
3	<b>LLO 3.1</b> Install and use external Python libraries using pip.	Install and demonstrate usage of any external library (e.g., requests).	CO1
4	<b>LLO 4.1</b> Develop programs using iteration, user-defined functions, and recursion.	Write recursive programs with proper logic implementation.	CO2
5	<b>LLO 5.1</b> Implement multi-level exception handling in programs.	Create programs demonstrating try, except, finally, custom exceptions.	CO2
6	<b>LLO 6.1</b> Perform read/write operations on CSV files.	Create programs to read, write and append data in CSV files.	CO2

7	<b>LLO 7.1</b> Perform serialization and deserialization using JSON.	Read and write JSON data using json module.	CO2
8	<b>LLO 8.1</b> Design basic GUI applications using tkinter widgets.	Create GUI with labels, buttons, entry fields.	CO3
9	<b>LLO 9.1</b> Implement event handling in GUI applications.	Develop a simple calculator or form with event handling and message box.	CO3
10	<b>LLO 10.1</b> Perform array creation and manipulation using NumPy.	Implement programs using NumPy arrays and mathematical operations.	CO4
11	<b>LLO 11.1</b> Manipulate and analyse tabular data using Panda	Read CSV file and perform filtering, sorting, grouping.	CO4
12	<b>LLO 12.1</b> Generate basic data visualizations using Matplotlib.	Create line, bar, and pie charts from sample data.	CO3,C04
13	<b>LLO 13.1</b> Visualize data from external files.	Create plots using data loaded from CSV files.	CO5
14	<b>LLO 14.1</b> Develop web applications using Flask framework.	Create Flask app with routing and form handling.	CO5
15	<b>LLO 15.1</b> Integrate database connectivity with Flask using SQLite.	Develop CRUD-based Student Record Web Application	CO5

## VI. SUGGESTED MICRO PROJECT/ASSIGNMENTS/ACTIVITIES FOR SELF LEARNING/SKILL DEVELOPMENT (SELF LEARNING)

### Micro Projects (Mini Applications / Use Cases)

1. Desktop Student Management System using tkinter and SQLite.
2. Marks Analysis Tool (read CSV, calculate average, generate bar chart).
3. Simple To-Do List Web App using Flask and SQLite.
4. Expense Tracker with GUI and file/CSV storage.
5. Library Book Management System (console or simple web).
6. Sales Data Visualizer (CSV input charts).
7. Basic Login System with Flask and database.
8. Simple Calculator GUI using tkinter.
9. CSV-based Employee Salary Calculator with charts.
10. Contact Book with SQLite storage.
11. Basic Login Page using Flask.

### Self-Learning / Skill Building Activities

1. Solve 30 medium-level problems on Hacker Rank/Python domain.
2. Create and upload projects to GitHub.
3. Download sample CSV datasets and practice Panda's operations.
4. Watch and replicate free YouTube tutorials on Flask/SQLite.
5. Prepare a report on "Real-world uses of Python in industry".
6. Record a demo video of your mini project.

## VIII. LIST OF INSTRUMENTS / EQUIPMENT / TRAINER BOARD

1	Visual Studio Code / PyCharm Community Edition / IDLE / Jupyter Notebook
2	Python Interpreter: Latest stable version of Python 3.x (from python.org)
3	<ul style="list-style-type: none"> <li>• Online Compilers &amp; Simulators</li> <li>• Replit: <a href="https://replit.com">https://replit.com</a></li> <li>• Programiz: <a href="https://www.programiz.com/python-programming/online-compiler/">https://www.programiz.com/python-programming/online-compiler/</a></li> <li>• Google Colab: <a href="https://colab.research.google.com">https://colab.research.google.com</a></li> </ul>
4	<ul style="list-style-type: none"> <li>• Collaboration / Documentation Tools:</li> </ul>

	<ul style="list-style-type: none"> <li>• GitHub / Git: Version control and collaboration</li> <li>• Google Docs / Microsoft Word: Report and assignment writing</li> <li>• Draw.io / Lucidchart: Flowcharts, diagrams, and logic design</li> </ul>
--	--

### IX. LIST OF REFERENCE BOOKS

Sr.No	Title	Author	Publication
1	Automate the Boring Stuff with Python (2 <sup>nd</sup> Edition)	Al Sweigart	No Starch Press, 2020
2	Python Crash Course (3 <sup>rd</sup> Edition)	Eric Matthes	No Starch Press, 2023
3	Head First Python (2 <sup>nd</sup> Edition)	Paul Barry	O'Reilly Media, 2016
4	Learning Python	Mark Lutz	O'Reilly Media (latest affordable)
5	Black Hat Python (for basic scripting)	Justin Seitz	No Starch Press

### X. LINK OF LEARNING WEB RESOURCE

1	<a href="https://automatetheboringstuff.com">https://automatetheboringstuff.com</a> — Practical Python projects
2	<a href="https://www.w3schools.com/python/">https://www.w3schools.com/python/</a> — Simple tutorials
3	<a href="https://realpython.com">https://realpython.com</a> — Beginner to intermediate articles
4	<a href="https://flask.palletsprojects.com/en/3.0.x/quickstart/">https://flask.palletsprojects.com/en/3.0.x/quickstart/</a> — Flask quickstart
5	<a href="https://www.sqlite.org/index.html">https://www.sqlite.org/index.html</a> — SQLite documentation

### XI. SUGGESTED WEIGHTAGE TO LEARNING EFFORTS & ASSESSMENT PURPOSE

Unit	Unit Title	Aligned COs	Learning Hours	R-Level	U-Level	A-Level	Total Marks
1	Advanced modules, exceptions, file handling.	CO1, CO2	12	4	4	4	12
2	GUI with tkinter	CO3	10	2	3	5	10
3	NumPy, Pandas, Matplotlib.	CO4	14	2	4	8	14
4	Flask, SQLite, integration	CO5	18	1	3	10	16
5	Mini Project	CO1, CO5	6	0	1	5	8
<b>Grand Total</b>			<b>60</b>	<b>9</b>	<b>15</b>	<b>32</b>	<b>60</b>

### XII. COs AND POs AND PSOs MAPPING

Course Outcome (Cos)	Programme Outcomes (POs)							Programme Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	2	1	0	0	0	1	0	1	0	1
CO2	3	2	3	2	0	2	0	3	0	2
CO3	3	3	3	3	0	3	0	3	0	3

CO4	3	2	3	3	0	2	0	3	0	2
CO5	3	3	3	3	3	3	2	3	0	3
<b>Legends:</b> - <i>3-High; 2-Moderate/Medium; 1-Slight/Low; 0-None</i>										