25CE104: Departmental Workshop (CE/IT/CSE)

w. e. f. Academic Year:	**2025-26**
Semester:	**1/2**
Category of the Course:	**Engineering Science**
Prerequisite:	Basic knowledge of computers and familiarity with using a keyboard, mouse, and common software applications.
Rationale:	This course introduces students to foundational programming concepts using Python through hands-on activities. It aims to develop logical thinking, problem-solving, and algorithmic skills from the first semester itself. The course emphasizes experiential learning, critical thinking and real-world applications. It lays a strong foundation for further studies in computer science and interdisciplinary computing.

Course Outcomes:

	Course Outcome (CO)	**RBT Level (Cognitive Domain)**
CO1	Apply programming concepts and logic-building skills to solve computational problems using Python.	Apply (Ap)
CO2	Develop structured and efficient Python programs using appropriate data types and control structures.	Apply (Ap)
CO3	Utilize built-in data structures and file handling techniques to store, retrieve, and manipulate data.	Apply (Ap)
CO4	Implement modular and reusable code by defining and using functions to enhance program readability.	Apply (Ap)
CO5	Demonstrate problem-solving and critical thinking skills by applying Python programming to real-life scenarios and case studies.	Analyze (An)

Teaching and Evaluation Scheme:

Teaching Scheme					**Examination Scheme**				
L	**T**	**P**	**C**	**Hrs/Week**	**IE**	**Theory**	**CIA**	**Practical**	**Total Marks**
-	-	02	01	02	-	-	30	20	50

IE: Internal Evaluation Theory: Theory Exam (End Semester)
CIA: Continuous Internal Assessment Practical: Practical Exam (End Semester)

List of Practical

S No	**Aim**	**Hours**
1.	A university department wants to automate student grade calculations. Taking input for marks across five subjects, using conditional logic to compute percentages and assign grades.	4
2.	A teacher wants to classify student roll numbers into two lists—odd and even. Lists, loops, and conditional statements.	2

3.	A store needs a system to manage product inventory. Using lists to store inventory items, tuples for fixed product details (like product codes and names), and operations such as adding, removing, and searching products.	4
4.	The college wants to maintain an alphabetically sorted list of student names for attendance. Lists, sorting methods (sorted () function), and user input handling.	4
5.	A library requires a system to store and retrieve book records. Reading from and writing to text files, managing book records (adding new entries, searching, and displaying all books).	4
6.	A weather website needs a tool to convert temperatures between Celsius, Fahrenheit, and Kelvin. Writing functions to perform unit conversions based on user input.	4
7.	A student wants to maintain a simple contact book to store names and phone numbers. Using dictionaries for data storage, adding, searching, and deleting contacts.8.	4
8.	A website needs to check the strength of a password based on certain criteria (length, uppercase, lowercase, digits, and special characters). Loops, conditional statements, and string functions.	4
	Total Hours	**30**

Text/Reference Books:

Textbooks:

1. **Python Programming:**
   - Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, 2nd Edition, 2023.
   - Paul Barry, *Head First Python*, O'Reilly Media, 2nd Edition, 2016.

Reference Books:

1. **Python Programming:**
   - Mark Lutz, *Learning Python*, O'Reilly Media, 5th Edition, 2013.
   - Zed A. Shaw, *Learn Python 3 the Hard Way*, Addison-Wesley, 1st Edition, 2017.
   - Al Sweigart, *Automate the Boring Stuff with Python*, No Starch Press, 2nd Edition, 2019.

Course Outcomes Mapping:

CO	Course Outcome	POs / PSOs	CL (Cognitive Level)	KC (Knowledge Category)	Class Sessions (Hours)
CO1	Apply programming concepts and logic-building skills to solve computational problems using Python.	PO1, PO2, PO12; PSO1	Apply (Level 3)	Procedural Knowledge	6
CO2	Develop structured and efficient Python programs using appropriate data types and control structures.	PO1, PO2; PSO1	Apply (Level 3)	Procedural Knowledge	6

CO3	Utilize built-in data structures and file handling techniques to store, retrieve, and manipulate data.	PO1, PO5; PSO1	Apply (Level 3)	Procedural Knowledge	6
CO4	Implement modular and reusable code by defining and using functions to enhance program readability.	PO1, PO2, PO11; PSO1	Apply (Level 3)	Conceptual Knowledge	6
CO5	Demonstrate problem-solving and critical thinking skills by applying Python programming to real-life scenarios and case studies.	PO2, PO3, PO12; PSO1, PSO2	Analyze (Level 4)	Conceptual & Procedural	6
Total hour					**30**

Mapping of COs with POs & PSOs:

CO	PO												PSO	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO1	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO2	3	3	-	-	-	-	-	-	-	-	-	-	3	-
CO3	3	-	-	-	2	-	-	-	-	-	-	-	3	-
CO4	3	2	-	-	-	-	-	-	-	-	2	-	3	-
CO5	-	3	2	-	-	-	-	-	-	-	-	3	3	2

3: High, 2: Medium, 1: Low