TYL – P3 Python
(Effective from the Academic Year 2025 -2026)
SEMESTER – IV

Course Code	**25TYLP3PY**	**CIE Marks**	**50**
Number of Contact Hours/Week	**2T + 2P**	**SEE Marks**	**50**
Total Number of Contact Hours	**40(Part A) + 20(Part B)**	**Exam Hours**	**1.5 Hours**

CREDITS –2

Course Learning Objectives: This course will enable the students to:

1. Learn the basics of Python programming and improve logical thinking.
2. Understand built-in data structures such as lists, tuples, sets, and dictionaries.
3. Work with strings and regular expressions for text handling and pattern matching.
4. Learn object-oriented programming concepts and write modular programs.
5. Understand fundamentals of algorithm design, searching, sorting, and DSA problem-solving.

Part - A
(Python Programming Foundations)

Module 1	**Contact Hours**
Review of Basic Programming & Functions: Quick recap of data types, Operators, variables, Input/Output, Branching, Loops **Advanced Functions:** Lambda functions, Map, Filter, Reduce, Recursion (advanced problems)	**10**
Module 2	
Lists: Review of list operations, list comprehensions, usage of built-in functions like zip(), enumerate(), all(), any(). **Tuples:** Packing, unpacking, and returning multiple values from functions. **Sets:** Creation, operations (union, intersection, difference, symmetric difference), and methods (add(), remove(), discard(), update()).	**7**
Module 3	
Strings and Regular Expressions: String manipulation in Python, String methods and functions. **Regular expressions:** Patterns matching using re.match(), re.search(), re.findall(), re.sub(). Solving string-related problems, Text Processing utility.	**7**
Module 4	
Dictionary: Dictionary comprehension, with conditions, defaultdict, OrderedDict, ChainMap, Counter (from collections), Dictionary-based problem solving (e.g., frequency map, anagram grouping, caching). **Iterators and Generators:** Iterator protocol (__iter__, __next__), generator functions using yield, generator expressions, use cases.	**8**
Module 5	

Object Oriented Programming Concepts: OOPs basics: Classes and Objects, Abstract method, Static Attributes, Methods, Counter, Inheritance, Overriding, super, types. **Review of DBMS**	**8**
Part - B **(DSA-driven Algorithmic Design and Problem-solving)**	
Module - 1	
Algorithm Design Concepts: Algorithm analysis and complexity (time and space).	**5**
Module - 2	
String Processing and Algorithms: **String Algorithms:** lexicographical sorting, common string problems – palindrome detection, anagram detection, substrings, string compression, **Assignment:** small projects involving string manipulation such as text-based games and data parsing.	**4**
Module-3	
2D List Operations: adding and modifying elements, sum and product of elements, matrix transpose **Dynamic 2D lists:** memory allocation for 2D lists, using pointers to create and manipulate 2D lists.	**4**
Module-4	
Searching Algorithms: linear search, binary search **Comparison Based sorting:** bubble sort, selection sort, insertion sort, merge sort, quick sort **Non-comparison Based sorting:** counting sort, radix sort.	**4**
Module-5	
Pattern Problems: star patterns using loops, Pascal's Triangle, Palindromic Pattern, etc. **Two Pointer Technique:** sorted pair sum, duplicates removal **Hash Maps in Python:** using dictionaries and collections. **Counter :** Fast lookup, Frequency mapping, Prefix sum problems **Assignment:** Solve key string/array problems using hash maps and pointers	**4**
Course Outcome: The student will be able to	
1. Write Python programs using functions, loops, and built-in features. 2. Use lists, tuples, sets, and dictionaries to solve real-time problems. 3. Perform string manipulation and apply regular expressions for pattern matching. 4. Implement object-oriented concepts and use iterators/generators efficiently. 5. Solve basic DSA problems using searching, sorting, two-pointer method, and hash maps.	
Question Paper Pattern and Evaluation Method:	
● **Part A:** Two coding questions – 20 marks and 30 marks	

- **Part B:** 10 MCQs (20 marks) and 1 coding question (30 marks)
- **Minimum passing marks:** 60%

Textbooks:

1. Al Sweigart, “Automate the Boring Stuff with Python”,1st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
2. Paul Barry, “Head First Python”,2nd Edition, O'Reilly Media, Inc,2016, ISBN: 9781491919538
3. Roberto Tamassia, Michael H Goldwasser, Michael T Goodrich, “Data Structures and Algorithms in Python”, 1st Edition, Wiley India Pvt Ltd, 2016. ISBN-13: 978- 8126562176

Reference Books and Web References:

1. Wesley J Chun, “Core Python Applications Programming”, 3rd Edition, Pearson Education India, 2015. ISBN-13: 978-9332555365
2. https://infytq.infosys.com/
3. https://www.geeksforgeeks.org/python-programming-examples/
4. https://www.geeksforgeeks.org/data-structures/
5. https://sites.google.com/cmrit.ac.in/p3python/home