

COURSE TITLE	PYTHON PROGRAMMING
COURSE CODE	01ML0101
COURSE CREDITS	2

Objective:

- 1 To develop fundamental programming skills in Python, implement data structures and algorithms for efficient problem-solving, and apply Python programming to address engineering challenges.

Course Outcomes: After completion of this course, student will be able to:

- 1 Understand Python Programs for Problem-Solving
- 2 Implement Control Structures and Functions in Python
- 3 Apply Data Structures and Algorithms for Efficient Computation
- 4 Utilize Python Libraries for Scientific Computing and Data Analysis
- 5 Solve Engineering and Mathematical Problems Using Python

Pre-requisite of course:NA

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
0	0	4	0	0	0	25	25
Contents : Unit	Topics						Contact Hours
Total Hours							

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	Implementation of basic python operations: variables, data types, and input-output handling Introduction to Python and its applications in engineering, Setting up the Python environment (Jupyter Notebook, Google Colab, IDLE), Writing and executing a basic Python program, Variables and data types (integers, floats, strings, booleans) Operators in Python (arithmetic, comparison, logical, bitwise), Input and output operations in Python.	4

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
2	Control flow in python: implementing conditional statements and looping constructs Conditional statements (if, elif, else) Looping structures (for loop, while loop, nested loops), Using break, continue, and pass statements, Functions in Python (defining, calling, parameters, return values)	4
3	Working with functions and file handling: creating, reading, and writing files in python Types of functions: Built-in functions, user-defined functions, lambda functions Recursion in Python, File handling (reading, writing, appending, handling different file types), Exception handling (try-except blocks, raising exceptions, multiple exceptions), Debugging techniques in Python	4
4	Understanding and Implementing Lists and Tuples in Python Lists Creating, indexing, slicing, adding, removing elements, Tuples: characteristics, indexing, iteration, immutability	4
5	Implementation of Dictionaries and Sets in Python Dictionaries: Key-value pairs, adding, updating, deleting elements, Sets: Creating, performing set operations (union, intersection, difference)	4
6	Introduction to Object-Oriented Programming (OOP) in Python Introduction to Object-Oriented Programming (OOP) – Classes, Objects, Methods, Concept of Inheritance and Polymorphism in Python	4
7	Implementation of Searching, Sorting Algorithms, and Object-Oriented Programming Concepts in Python Searching algorithms: Linear Search, Binary Search, Sorting algorithms: Bubble Sort, Selection Sort, Insertion Sort,	4
8	Numerical computations using numpy: performing matrix operations and solving linear equations in python Introduction to NumPy: Arrays, indexing, slicing, reshaping, Matrix operations using NumPy: Addition, multiplication, inverse, transpose	4
9	Introduction to Data Visualization using Matplotlib in Python Introduction to Matplotlib: Line plots, bar charts, scatter plots, pie charts, Customizing graphs with Matplotlib: Labels, legends, grid, colors,	4
10	Introduction to Data Analysis using Pandas and Data Visualization with Matplotlib Introduction to Pandas: DataFrames, Series, handling missing data, Reading and writing CSV and Excel files using Pandas, Data filtering and transformation using Pandas, Data visualization using Pandas and Matplotlib	4

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
11	Application of Python for Differential and Integral Calculus Differentiation of functions using SymPy, Integration of functions and definite integrals, solving differential equations using Python	4
12	Application of Python for Elements of Mechanical Engineering Solving equations for thermodynamic cycles (Carnot, Otto, Diesel), Work and heat transfer calculations in thermodynamics, Solving simple mechanical force balance equations	4
Total Hours		48

Textbook :

- 1 The complete reference Python, Martin C brown , TMH, 2018
- 2 Python Programming using problem-solving approach, Reema Thareja, Oxford Publications, 2023

References:

- 1 Introduction to Python Programming , Introduction to Python Programming , Gowrishankar S, Veena A, Routledge, 2018
- 2 Core Python Programming , Core Python Programming , Dr. R. Nageswara Rao, Wiley, 2018
- 3 Python Programming, Python Programming, Dr. Jisu Elsa Jacob, Bharath Viswam S, S.K. Kataria & Sons, 2022
- 4 Python Crash Course , Python Crash Course , Eric Matthes, No Starch Press, 2019
- 5 Automate the Boring Stuff with Python , Automate the Boring Stuff with Python , Al Sweigart, No Starch Press, 2020

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember / Knowledge	Understand	Apply	Analyze	Evaluate	Higher order Thinking / Creative
10.00	30.00	20.00	20.00	20.00	0.00

Instructional Method:

- 1 Presentation
- 2 Programming with Python

Supplementary Resources:

- 1 <https://nptel.ac.in/courses/106106145>