

Semester	2
Course	Major
Paper Code	C1DS250211P
Paper Title	Python Programming
No. of Credits	4
Theory/Composite/ Practical	Practical
Minimum no. of preparatory hours per week a student has to devote	5
Number of Modules	1
Syllabus	Introduction to Python [2L] Applications; technical strengths; Python programming environments; program execution in Python. Python operators and statements [12L] Various operators in Python – assignment, arithmetic, relational, logical and bitwise; Sequential statements; importance of indentation; if statement and its variations; loops – for and while; keywords used with loops – break, continue, pass, Loop else. Python types and operations [14L] Numeric types; String; List; Dictionary; Tuple; Set. Functions in Python [14L] The def keyword; function scope; function arguments; recursive functions; anonymous functions – the lambda keyword; functional programming tools – filter, map and reduce. Modules and Packages [4L] Need for modules; working of imports; module creation; module usage; module namespaces; package imports; need for package imports. Object-Oriented Programming (OOP) in Python [14L] OOP principles and their implementation in Python – data hiding, encapsulation, abstraction, polymorphism and inheritance; exception handling. File handling [6L] Opening a file; file modes; reading a file; writing to a file; closing a file; file handling exceptions. Supporting libraries in Python [6L] Basic numpy and pandas processing of data
Learning Outcomes	• Understanding the fundamentals of programming such as algorithms and flowcharts. • Appreciating the benefits of the Python programming language by gaining insights into how a Python program is executed. • Writing efficient programs by making use of Python programming constructs and powerful built-in data structures. • Designing classes in Python from scratch by defining user-defined functions and implementing object-oriented programming (OOP) principles.

	• Gaining a thorough understanding of functional programming techniques using <code>lambda</code>, <code>map</code>, <code>filter</code>, and <code>reduce</code>. • Handling various types of files using libraries such as NumPy and pandas, and managing exceptions that may arise during file operations.
Reading/Reference List	1. Python for Everybody. Author: Charles R. Severance. Publisher: Shroff Publishers. 2. Learning Python. Author: Mark Lutz. Publisher: O'REILLY. 3. Python Programming: Using Problem Solving Approach. Author: Reema Thareja. Publisher: OXFORD.
Evaluation	Continuous Practical Assessments