COURSE TITLE	**PYTHON PROGRAMMING**
COURSE CODE	**05FN0306**
COURSE CREDITS	**4**

Objective:

1 Build a strong foundation in Python syntax, control structures, functions, and file handling.
2 Apply OOP concepts like classes, objects, inheritance, and polymorphism in Python.
3 Develop modular, reusable, and efficient code using Python's OOP features.
4 Gain hands-on experience through projects and case studies.
5 Build industry-ready applications with GUI and automation.

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

1 Write and execute Python programs using loops, functions, and file handling.
2 Implement object-oriented concepts like classes, objects, inheritance, and polymorphism.
3 Develop modular and reusable Python code for real-world applications.
4 Use file handling and exception handling techniques efficiently.
5 Build GUI-based and database-driven applications using Python.

Pre-requisite of course:NA

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	2	0	30	20	50	50

Contents : Unit	Topics	Contact Hours
1	**Introduction to Python** Overview of Python,Installation and Setup (IDLE, Jupyter, VS Code),Python Syntax, Keywords, and Identifiers, Data Types and Type,Conversion Variables and Operators,Control Flow Statements, Conditional Statements (if, elif, else),Looping Constructs (for, while),Iterators and Generators	11
2	**Functions and Modules** Defining and Calling Functions,Function Arguments and Return Values,Lambda Functions, Recursion,Modules and Packages,Built-in Functions and Libraries,File Handling (Reading, Writing, Appending Files)	11

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
3	**Object-Oriented Programming in Python** Introduction to OOPs,Classes and Objects,Constructors (__init__ method),Instance, Class, and Static Methods, Encapsulation (Private and Public Attributes),Inheritance (Single, Multiple, Multilevel),Polymorphism (Method Overriding, Operator Overloading),Abstraction (Abstract Classes and Interfaces)	11
4	**Advanced OOPs and Real-world Applications** Exception Handling (try, except, finally),File Handling with OOP,Magic Methods (__str__, __repr__, __len__, etc.), Decorators and Property Methods,Introduction to GUI (Tkinter, PyQt), Mini Project: Implementing OOP Concepts in a Real-world Application	12
Total Hours		**45**

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Unit-I** Write a Python program to perform basic arithmetic operations. , Implement a Python script to swap two numbers without using a third variable., Develop a program to check whether a given number is prime or not., Write a Python script to find the largest of three numbers using conditional statements., Implement a menu-driven program for basic mathematical operations using functions.	15
2	**Unit-II** Implement a menu-driven program for basic mathematical operations using functions., Develop a Python script to demonstrate different types of function arguments (default, keyword, and variable-length). , Implement a Python program to read and write data from a text file., Create a Python script to count the occurrences of words in a file., Develop a program to demonstrate exception handling in file operations.	15
3	**Unit-III** Write a Python program to create a class with attributes and methods. , Implement inheritance by creating a parent and child class with overridden methods., Develop a Python script to demonstrate method overloading and method overriding., Create a Python program to implement encapsulation using getter and setter methods., Implement a class-based approach to calculate the area of different geometric shapes.	15

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
4	**Unit-IV** Write a Python program to implement operator overloading using magic methods, Develop a Python script to demonstrate multiple inheritance and method resolution order (MRO)., Implement a simple GUI application using Tkinter for a calculator., Create a Python program to connect to a database and perform CRUD operations. , Develop an OOP-based application for a simple employee management system.	15
Total Hours		**60**

Textbook :

1 Learning Python, Mark Lutz, O'Reilly Media, 2013

2 Python Crash Course, Eric Matthes, No Starch Press, 2019

References:

1 Automate the Boring Stuff with Python, Automate the Boring Stuff with Python, Al Sweigart, No Starch Press, 2020

2 Fluent Python, Fluent Python, Luciano Ramalho, O'Reilly Media, 2022

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					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
20.00	30.00	25.00	15.00	10.00	0.00

Instructional Method:

1 Board Work,PPT,Practicals

Supplementary Resources:

1 Python Official Documentation" – https://docs.python.org/3/

2 GeeksforGeeks Python Tutorials – https://www.geeksforgeeks.org/python-programming-language/

3 https://docs.python.org/3/tutorial/index.html

4 https://docs.python.org/3/tutorial/index.html

5 https://www.tutorialspoint.com/python/index.htm

6 https://www.learnpython.org/

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)