

25UCST403	PROGRAMMING IN PYTHON	Category	L	T	P	R	Credit	Year of Introduction
		PBL	3	0	0	1	4	2025

Preamble: The course is to provide learners an insight into python programming to develop multi-module software solutions for real-world computational problems. It covers the Python programming environment, language syntax, data representation, intermediate programming constructs, GUI development, object-oriented programming, and data processing techniques. The course provides a strong foundation for building modular software systems, including complex interactive applications, network-based solutions, and data-driven intelligent applications.

Prerequisite: Nil

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

CO	Course Outcomes	Knowledge Level
CO1	Apply Python fundamentals, control structures, data structures, and functions to develop modular programs.	K3 Apply
CO2	Implement Python applications using object-oriented principles and file operations to produce modular and reliable software solutions.	K3 Apply
CO3	Apply Python libraries to build graphical outputs, create event-driven interfaces, and manage persistent data using text, CSV, and serialization.	K3 Apply
CO4	Employ NumPy, Pandas, and plotting tools for data processing, analysis, and visualization for decision support.	K3 Apply

Mapping of course outcomes with program outcomes and program specific outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1	3	2	1	-	2	-	-	-	1	-	1	1	1
CO2	2	1	1	-	2	-	-	-	2	-	1	1	2
CO3	2	1	2	-	2	-	-	-	2	-	1	1	2
CO4	3	2	-	2	3	-	-	-	2	-	1	1	2
Avg	2.5	1.5	1.33	2	2.25	-	-	-	1.75	-	1	1	1.75

Mapping of course outcomes with Knowledge and Attitude Profile (WK)

	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	*	*	*	*	*	*		*	
CO2	*	*	*	*	*	*		*	
CO3	*	*	*	*	*	*		*	
CO4	*	*	*	*		*		*	

Wks Mapped with the course

WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
*	*	*	*	*	*		*	

Mark Distribution:

Total Marks	CIE Marks	ESE Marks	ESE Duration
100	60	40	2.5 hrs

Continuous Internal Evaluation Pattern:

Attendance	: 5 marks
Project	: 30 marks
Continuous Assessment Test	: 25 marks

PBL Course Elements

L: Lecture (3 Hrs.))	R: Project (1 Hr.), 2 Faculty Members		
	Tutorial	Practical	Presentation
Lecture delivery	Project identification	Simulation/Laboratory Work/Workshops	Presentation (Progress)
Group discussion	Project Analysis	Data Collection	Evaluation
Question Answer Sessions	Analytical thinking and self-learning	Testing	Project Milestone Reviews, Feedback.
Guest Speakers (Industry Experts)	Case Study/ Field Survey Report	Prototyping	Presentation: Students present their results.

Assessment and Evaluation for Project Activity

Sl.No	Evaluation Rubrics	Allotted Marks
1	Problem Definition & Design	5

2	Implementation Using OOP	5
3	Use of Python Features	6
4	Code Quality	4
5	Documentation (Industrial visit/training report)	5
6	Demo & Viva	5
Total		30

1. Problem Definition & Design (5 Marks)

1. Clear statement of the problem and objectives
2. Proper requirement analysis (inputs, outputs, constraints)
3. Suitable identification of classes, objects, attributes, and relationships
4. Use of design representations (class diagram / flowchart / module diagram)

5. Implementation Using OOP in Python (5 Marks)

1. Correct application of OOP principles
2. Proper creation and usage of classes and objects
3. Meaningful data members and methods
4. Logical structure following object-oriented design

5. Use of Python Features (6 Marks)

1. Constructors, destructors where needed
2. Inheritance, polymorphism, encapsulation, abstraction
3. Method overloading alternatives / operator overloading
4. Use of libraries, exception handling, file handling, modules

5. Code Quality (4 Marks)

1. Readable and modular code
2. Proper naming conventions (PEP style)
3. Indentation, comments, formatting
4. Efficient logic with minimal repetition

5. Documentation (5 Marks)

1. Introduction and objectives
2. Design details and class descriptions
3. Algorithm / module explanation
4. Sample input/output, screenshots

5. Demo & Viva (5 Marks)

1. clearly explain the logic and working of the program.
2. demonstrate understanding of the classes, objects, and OOP principles used.
3. executed successfully with valid test cases.
4. Communication & confidence

SYLLABUS

Module 1 (14 hrs)

Getting started with Python programming- Running code in the interactive shell- Editing, Saving and Running a script-Usage of Editors and IDE.Python identifiers -variables- keywords- operators- operator precedence- indentation. Python program flow control statements-decision making statements- Loops. Built-in data structures: lists, tuples, sets, dictionaries, and strings-basic searching and sorting-Function design and usage, parameter passing, return values, scope, recursion, lambda expressions, higher-order functions, modules and packages.

Module 2 (14 hrs)

Defining classes-creating objects-instance variables and class variables-methods-constructors -accessors and mutators-encapsulation-data hiding conventions-structuring classes with inheritance and polymorphism- Abstract classes- Exception handling and file handling basics: Opening and closing files- file modes- reading and writing text files-working with structured data.

Module 3 (14 hrs)

Graphics–Terminal-based programs, Simple Graphics using Turtle, Operations, 2D Shapes-3D Shapes-Colors. Graphical User Interfaces- Event-driven programming-Coding simple GUI Programs- windows, labels, image display, text input, dialog boxes, command buttons, with case studies. File Processing: Reading and Writing Text Files-Data Manipulation and Analysis- CSV file handling.

Module 4 (14 hrs)

NumPy: Basics-creating arrays- Arithmetic-slicing-Matrix operations, and random numbers-Linear Algebra-Solving Non-Linear Equation. Pandas: Reading-Manipulating and Processing Data-Data Frame -Grouping and Aggregation. Visualization with Matplotlib: creating basic plots-customizing ticks- labels-legends and presenting graphical representations of data.API handling in python.

Completion of an industrial visit or industrial training is desirable for this course.

Text Books:

5. Lambert, K. A. (2023). *Fundamentals of Python: First programs* (3rd ed.). Cengage Learning.
6. McKinney, W. (2022). *Python for data analysis: Data wrangling with pandas, NumPy, and Jupyter* (3rd ed.). O'Reilly Media.

Reference Books:

7. Chun, W. J. (2006). *Core Python programming* (2nd ed.). Prentice Hall.
8. Miller, B. N., & Ranum, D. L. (2011). *Problem solving with algorithms and data structures using Python* (1st ed.). Franklin, Beedle & Associates.
9. Matthes, E. (2023). *Python crash course: A hands-on, project-based introduction to programming* (3rd ed.). No Starch Press.

Suggestion on Project Topics

Students must identify and implement a project using Python with Object Oriented Programming (OOP) concepts. The selected topic should satisfy the following conditions.

Project Nature Requirements

1. User Data Input Requirement

The project must:

1. Accept a significant amount of input from users.
2. Provide a menu-driven interface.
3. Use classes and objects for storing and managing data.
4. Validate input (type checking, range checking, format).

5. Permanent Data Storage Requirement

The project must permanently store data using:

1. File handling (text / CSV / JSON / binary using pickle)
2. Simple database usage (optional extra).

Data organization must be done through **class structures**, not loose variables.

3. Data Processing Requirements

The project must process both:

1. User-entered data
2. Previously stored file/database data to produce meaningful outputs such as:
 1. Reports
 2. Summaries
 3. Searching & filtering
 4. Calculations
 5. Statistics or analysis

Outputs must be displayed clearly to the user.

Examples

Vehicle Service Management: Develop a system to maintain customer and vehicle information, record service history, calculate service charges, and retrieve previous maintenance details using file storage.

Travel Booking System: Create a booking system that allows passengers to reserve tickets, check availability, calculate fares, and store booking information for future reference.

Shop Management System: Implement a solution to manage product details, handle sales transactions, update stock levels, and generate daily or monthly reports.

Course Code	25UCST403
Course Name	PROGRAMMING IN PYTHON
Max. Marks: 40	Duration: 2.5 Hours

Part A <i>(Answer all questions. Each question carries 3 marks)</i>				
Qn No.	Question	Marks	CO	BL
1	A student writes print("Welcome") in the interactive shell. Apply the command and state the expected outcome.	2	1	L3
2	Determining the number of possible arrangements of inventory items requires factorial evaluation. Develop a recursive Python program to perform the task.	2	1	L3
3	Shipment datasets maintained in CSV format require structured data processing for report generation. Determine the Python facility that can be employed for the operation.	2	2	L3
4	Customer information systems require customer data to be accessed and modified only through designated methods. Implement encapsulation to control data access.	2	2	L3
5	Input validation failures require immediate notification to the user. Implement a dialog box to present the warning message.	2	3	L3
6	Sensor readings collected from an industrial monitoring unit need to be preserved for subsequent analysis. Implement suitable file operations to store the readings.	2	3	L3
7	Traffic flow data represented as matrices requires computational analysis. Employ matrix multiplication operations to process the data.	2	4	L3
8	Customer purchase data requires classification based on region and category. Utilize grouping functionality to organize the records.	2	4	L3
Part B <i>(Answer any one full question from each Module. Each carry 9 marks)</i>				
Module I				
Qn No.	Question	Marks	CO	BL
9	Digital library stores book information and allows searching by title. Construct a Python program using dictionaries and searching techniques to retrieve book details.	6	1	L3
OR				
10	A food delivery application applies a discount percentage to all order amounts. Use a lambda expression with a higher-	6	1	L3

	order function to implement the solution.			
Module II				
Qn No.	Question	Marks	CO	BL
11	Smart city transportation system maintains common attributes such as vehicle ID, route number, and fare details for all transport services. Metro trains and electric buses require additional attributes specific to their operations. Construct an inheritance hierarchy to model the system and demonstrate its usage.	6	2	L3
OR				
12	A multinational organization maintains payroll records for permanent, contract, and consultant employees, each requiring distinct salary computation rules. Develop a solution that utilizes polymorphism to process salary calculations and file handling techniques to store and retrieve payroll information	6	2	L3
Module III				
Qn No.	Question	Marks	CO	BL
13	Multicolored polygon-based designs are to be generated in a computer-aided design environment for graphical visualization. Implement a Turtle graphics program to create the required pattern.	6	3	L3
OR				
14	Appointment scheduling requires users to enter personal details, select options, and receive confirmation messages through graphical controls. Develop a GUI application using suitable widgets and events.	6	3	L3
Module IV				
Qn No.	Question	Marks	CO	BL
15.a	Integrate shipment tracking data from a cloud-based API into a Python application for monitoring purposes.	3	4	L3
15.b	Generate a graphical report for rainfall measurements collected over several months using Matplotlib.	3	4	L3
OR				
16.a	Implement a Python solution that accesses a web API and processes JSON-formatted response data.	3	4	L3
16.b	Process a dataset containing student marks using Pandas and generate department-wise average performance statistics through grouping and aggregation operations.	3	4	L3
