LESSON PLAN 2025-26(WINTER)

NAME OF THE TEACHER : DEEPAK KUMAR BARDA, LECT.(STAGE-II,CSE)

Subject: **PROGRAMMING WITH PYTHON (Course Code: CSEPC 203/TH2)**
Program: Diploma in Computer Science and Engineering
Semester: 3rd
Total Contact Hours: 45
Total Marks: 100
Assessment: Internal Assessment – 30, End Term – 70

After completion of the course, the students will be able to:
CO1-Define Python's core syntax, data types, and key concepts of object-oriented programming.
CO2-Explain how control structures and data structures function in Python.
CO3-Implement Python programs using file handling, modules, and libraries like NumPy, Pandas, and Matplotlib for data analysis and visualization.
CO4- Analyze Python scripts to identify and resolve logical or syntactic errors and optimize code using advanced techniques like recursion and lambda functions.
CO5-Develop a real-world mini-project by integrating Python concepts such as OOP, libraries, and automation tools for practical problem-solving.

Day	Unit	Topic	Learning Objective	Activities	Homework	COURSE OBJECTIVE
Unit I: Introduction to Python (8 Lectures)						
1	I	Overview of Python: Features and Applications	Understand what Python is and its main uses in various fields.	Discussion on Python's versatility; show examples of Python in web dev, data science, and automation.	Research a famous company that uses Python and write a short summary of how they use it.	CO1
2	I	Setting Up the Python Environment	Install Python and a code editor (like VS Code or PyCharm).	Guided installation of Python and a chosen IDE. Run a simple "Hello, World!" script.	Set up the environment on your own computer and run the "Hello, World!" script.	CO1
3	I	Python Syntax: Variables and Data Types	Define and use variables; identify different data types (e.g., int, float, str).	Interactive coding session with variable assignment and type checking using type().	Write a script that declares variables for your name, age, and favorite number.	CO1
4	I	Python Operators	Use arithmetic, comparison, and logical operators.	Whiteboard exercise to solve simple mathematical and logical problems using Python operators.	Create a program that converts Celsius to Fahrenheit using arithmetic operators.	CO1
5	I	Writing, Executing, and Debugging Scripts (1/2)	Write and save basic Python scripts.	Step-by-step walkthrough of writing a script, saving it, and executing it from the terminal.	Write a small script to calculate the area of a rectangle.	CO1

6	I	Writing, Executing, and Debugging Scripts (2/2)	Understand and fix common syntax and runtime errors.	Introduce common errors (e.g., SyntaxError, NameError). Students intentionally make errors and debug.	Debug a provided Python script that contains multiple errors.	CO1
7	I	Review and Mini-Project	Consolidate knowledge of Unit I through a practical exercise.	Develop a simple command-line calculator that takes two numbers and an operator as input.	Finish the calculator project and add a "power" operator.	CO1
8	I	Unit I Assessment	Demonstrate mastery of Unit I topics.	Written quiz on concepts and a short practical coding test.	N/A	CO1
Unit II: Control Structures and Functions (8 Lectures)						
9	II	Conditional Statements: if, else, elif	Use conditional statements to control program flow.	Live coding a simple program that checks if a number is positive, negative, or zero.	Write a script that determines if a person is old enough to vote.	CO2
10	II	Loops: for loop	Iterate over sequences using a for loop.	Practice using for loops with lists, strings, and the range() function.	Write a program to print the first 10 even numbers.	CO2
11	II	Loops: while loop	Use a while loop for indefinite iteration.	Compare for and while loops. Write a simple guessing game using a while loop.	Create a program that prompts the user for a password until they enter the correct one.	CO2
12	II	Nested Loops	Use one loop inside another to solve complex problems.	Draw flowcharts for nested loops before coding; create a program that prints a pattern of asterisks.	Write a program that generates a multiplication table up to 10x10.	CO2
13	II	Functions: Defining and Calling	Create and call your own functions.	Practice defining functions with and without parameters and return values.	Convert your rectangle area program into a function.	CO2
14	II	Functions: Scope of Variables	Understand local and global variable scope.	Explain the concept of scope with a visual diagram. Code examples that demonstrate variable scope rules.	Predict the output of provided code snippets that involve different variable scopes.	CO2

15	II	Introduction to Lambda Functions	Write simple, anonymous functions.	Show how to use lambda functions with filter() and map().	Convert a simple named function into a lambda function.	CO2
16	II	Recursion	Solve problems by calling a function from within itself.	Walk through the factorial and Fibonacci sequence examples, emphasizing the base case.	Write a recursive function to calculate the sum of numbers from 1 to N.	CO2

Unit III: Data Structures in Python (9 Lectures)						
17	III	Data Structures: Lists (1/2)	Create and manipulate lists.	In-class exercises on list creation, indexing, slicing, and common methods like append(), remove().	Write a program to find the largest number in a list.	CO3
18	III	Data Structures: Lists (2/2)	Work with lists of different data types and nested lists.	Practice list operations like concatenation and sorting.	Create a list of your favorite movies and then sort them alphabetically.	CO3
19	III	Data Structures: Tuples	Understand the immutability of tuples and when to use them.	Compare lists and tuples; practice tuple packing and unpacking.	Create a function that returns a tuple of the average and sum of a list of numbers.	CO3
20	III	Data Structures: Sets	Use sets for unique element storage and mathematical operations.	Code examples demonstrating how to add/remove elements and perform set operations (union, intersection).	Given two lists, find the common elements and the unique elements in each.	CO3
21	III	Data Structures: Dictionaries (1/2)	Create and access key-value pairs in dictionaries.	Build a simple contact book or glossary using dictionaries.	Create a dictionary of countries and their capitals.	CO3
22	III	Data Structures: Dictionaries (2/2)	Iterate through dictionaries and use common methods.	Practice using .keys(), .values(), and .items() to loop through a dictionary.	Write a program to count the frequency of each word in a given sentence.	CO3
23	III	List Comprehensions and Dictionary Comprehensions	Write concise code for creating new lists and dictionaries.	Convert traditional loops for list and dictionary creation into one-line comprehensions.	Rewrite your list of even numbers program using a list comprehension.	CO3
24	III	Working with Strings	Manipulate strings using methods and slicing.	Practice with string methods like upper(), lower(), strip(), and split().	Write a function that takes a sentence and returns a list of its words.	CO3
25	III	Python's collections Module	Use specialized container data types.	Introduce Counter, defaultdict, and deque with practical use cases.	Use Counter to find the most common word in a paragraph.	CO3
Unit IV: File Handling and Modules (5 Lectures)						
26	IV	File Operations: Reading, Writing, Appending	Read from and write to text files.	Hands-on practice opening, reading, and writing to a .txt file. Emphasize using the with statement.	Write a program that reads a file and writes its content to another file.	CO4

27	IV	Working with CSV Files	Read and write data in CSV format.	Introduce the csv module. Read data from a sample CSV file and write data to a new one.	Create a program that reads a CSV file of student grades and calculates the average for each student.	CO4
28	IV	Working with JSON Files	Handle JSON data in Python.	Introduce the json module. Load data from a JSON file and dump Python data to a new JSON file.	Write a script to fetch data from a public API endpoint and save it to a local JSON file.	CO4
29	IV	Built-In Modules (e.g., math, os, datetime)	Use popular built-in modules to solve common problems.	Explore functions from math and os. Use datetime to get the current date and time.	Use the os module to list all files in a specific directory.	CO4
30	IV	Creating and Using Custom Modules	Organize code into reusable modules.	Show how to create a .py file and import its functions into another script.	Create a module with functions for common geometric calculations and use it in a separate program.	CO4
Unit V: Object-Oriented Programming (OOP) (5 Lectures)						
31	V	OOP: Understanding Classes and Objects	Define classes and create objects.	Walkthrough the concept of a class as a blueprint and objects as instances. Create a Dog class.	Create a Car class with attributes like color and make.	CO5
32	V	Concepts of Encapsulation and Inheritance	Understand how to hide data and create specialized classes.	Show how to use private attributes and demonstrate inheritance by creating a Poodle class that inherits from Dog.	Create a Vehicle class and then create an ElectricCar class that inherits from it.	CO5
33	V	Concepts of Polymorphism	Use a single interface for different data types.	Explain polymorphism through method overriding in subclasses.	Demonstrate polymorphism with a group of different animal classes that all have a speak() method.	CO5
34	V	Working with Magic Methods	Use dunder (magic) methods for operator overloading.	Show how __init__, __str__, and __add__ can customize class behavior.	Implement __str__ for your Car class to print a readable representation.	CO5

35	V	Exception Handling in Python	Handle errors gracefully using try, except, finally.	Introduce common exceptions and show how to use a try-except block to prevent a program from crashing.	Write a program that asks for a number but handles a ValueError if the user enters text.	CO5
Unit VI: Advanced Python and Applications (10 Lectures)						
36	VI	Introduction to Libraries: NumPy	Use NumPy for numerical operations on arrays.	Show the speed and efficiency of NumPy arrays compared to Python lists for mathematical operations.	Perform element-wise addition and multiplication on two NumPy arrays.	CO5
37	VI	Introduction to Libraries: Pandas	Use Pandas for data manipulation and analysis.	Introduce DataFrames and Series. Load data from a CSV file into a DataFrame and perform basic operations.	Calculate the mean and standard deviation of a column in a Pandas DataFrame.	CO5
38	VI	Mini-Project: Problem Selection	Choose a real-world problem to solve with Python.	Brainstorm project ideas and narrow down the scope. Start outlining the project requirements.	Write a brief project proposal detailing the problem, data source, and intended functionality.	CO5
39	VI	Mini-Project: Data Collection & Cleaning	Gather and prepare the data for the project.	Find a dataset online (e.g., Kaggle, data.gov) or create a mock dataset. Use Python to clean and preprocess the data.	Clean up the project's dataset, handling missing values and incorrect data types.	CO5
40	VI	Mini-Project: Core Logic Development (1/2)	Begin implementing the main logic of the project.	Write the core functions and classes for the project. Test each component individually.	Implement the first major feature of your mini-project.	CO5
41	VI	Mini-Project: Core Logic Development (2/2)	Continue building the project's functionality.	Integrate the different parts of the project. Work on any remaining features.	Finish implementing the core logic and ensure it runs without errors.	CO5
42	VI	Mini-Project: Refinement and Documentation	Improve code readability and add comments.	Review and refactor the code. Write clear comments and a README.md file explaining the project.	Document your code thoroughly with comments and a project overview.	CO5

43	VI	Mini-Project: Finalization and Presentation	Prepare to present the completed project.	Practice presenting the project to a peer. Finalize any last-minute changes.	Create a brief presentation summarizing your mini-project's purpose, features, and challenges.	CO5
44	VI	Mini-Project: Project Demonstration	Present your project to the class.	Each student presents their project, explaining the code and demonstrating its functionality.	N/A	CO5
45	Final Review	Comprehensive Course Review	Review all key topics from the entire course.	Q&A session covering all units. Work through a challenging problem that requires integrating multiple concepts.	N/A	CO5

Deepak Kumar Barde
11.07.2025
Signature of Teacher

Deepak Kumar Barde
11.07.2025
Signature of HOD