Semester-IV

Course code: CSE21VSP253 **Course name:** Adv. Python Programming **Course category:** VSEC
Credits: 2 Teaching **scheme:** P-4 hrs/week **Evaluation scheme:** TW–30, PR–20

Pre-requisites: Basic Python programming

.

Course Objectives:

1. To develop application using advanced python programming concepts.
2. To write object oriented programs with python classes.

Course Outcomes: At the end of the course, the students will be able to -

CO1: Implement python programs using the python data structures like lists, dictionaries, tuples, strings and sets.
CO2: Design object oriented programs with Python classes.
CO3: Familiarize the handling of I/O Exceptions and usage of Directories
CO4: Develop a program to perform basic SQL CRUD operations in python.
CO5: Build application using GUI and web programming in python

Contents –

Unit	Content
1	**Introduction to Python:** Basic coding skills, working with data types and variables, numeric data, string data, Python functions, Boolean expressions, selection structure, iteration structure, working with lists, list of lists, tuples, dates and times, get started with dictionaries
2	**Object Oriented Python Programming**: OOPS Concepts, Classes and objects , Classes in Python, Constructors, Data hiding, Creating Classes, Instance Methods, Special Methods, Class Variables, Inheritance, Polymorphism, Type Identification, Custom Exception Classes, Iterators, generators and decorators..
3	**I/O and Error Handling In Python :**Introduction, Data Streams, Creating Your Own Data Streams, Access Modes, Writing Data to a File, Reading Data From a File, Additional File Methods, Handling IO Exceptions, Errors, Run Time Errors, The Exception Model, Exception Hierarchy, Handling Multiple Exceptions, Working with Directories.
4	**An Introduction to relational databases:** SQL statements for data manipulation, Introduction to My SQL, PYMYSQL Connections, Using SQLite Manager to work with a database, Using Python to work with a database, Creating a GUI that handles an event, working with components. **Image processing**: crop, scale, rotate, flip, change contrast, brightness, color, edge detection, blur sharpening **GUI and Web Programming in python: Introduction to GUI**: Widget- Button, canvas, frame, label, listbox, menubutton, menu, messages, spinbox, paned window, Basic. Web Programming: JSON and XML, Using XML-RPC, Rest Interfaces, WSGI and HTML, Flask Framework, Controller Functions, templates and Forms, Database ORMs

Text Books:

1. Sakis Kasampalis Quan Nguyen Dr Gabriele Lanaro Dr. Gabriele Lanaro, Advanced Python Programming: Build high performance, concurrent, and multi-threaded apps with Python using proven design patterns
2. Michael H Goldwasser, David Letscher, “Object Oriented Programming in Python”, Prentice Hall, 1st Edition, 2007
3. Yashavant Kanetkar, Aditya Kanetkar, “Let us Python, BPB publication, 1st Edition, 2019.
4. Ashok Kamthane, Amit Kamthane, “Programming and Problem solving with Python”, McGraw Hill Education (India) Private Limited, 2018.

Reference Books:

1. R Nageswara Rao, "Core Python Programming", Dreamtech Press, 2017 Edition.
2. Taneja Sheetal, Kumar Naveen, "Python Programming – A Modular Approach", Pearson, 2017.

Suggestive List of Practical:

1. Write a program to count the numbers of characters in the string and store them in a dictionary data structure. Write a program to use split and join methods in the string and trace a birthday with dictionary data structure.
2. Write a program to perform various operations on matrices.
3. Write a function to find mean, median, mode for the given set of numbers in a list.
4. Write a function nearly equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
5. Demonstrate Basic date and time classes, Different time formats, Converting between formats, Formatting dates and times, Parsing date/time information
6. Create a class ATM and define ATM operations to create account, deposit, check_balance, withdraw and delete account. Use a constructor to initialize members.
7. a. Make a class Employee with a name and salary. Make a class Manager inherit from an employee. Add an instance variable, named department. Write a method that prints the manager name, department and salary. Make a class Executive inherit from Manager. Write a method that prints the string followed by the information stored in the Manager super class object.

7 b. Write a program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.

8. Write a program to create a class circle, square, triangle and rectangle. find the area of each objecting each class by applying the concept of polymorphism.
9. Read two numbers n1 and n2. Write a function to compute n1/n2 and use try/except to catch the exceptions.
10. Write a Python program to detect and handle the exception while solving the quadratic equation.
11. Write a Python program to handle the run time errors while doing file handling operation.
12. Write a python program to perform various database operations (creates, insert, delete, update).
13. Write a programs for performing the image processing operations
14. Display an Image and its Transformation using Tkinter and Open CV-Python
15. Design a GUI based calculator to perform arithmetic operations like addition, subtraction, multiplication and division. (Hint: Expression Calculator using tk)
16. Design a GUI based application to convert temperature from Celsius to Fahrenheit
17. Design a GUI based application to calculate Average Speed
18. Write a program for creating Django Application
19. Write a program for querying data with Django
20. Mini-project