BCSC 0860: COMPUTER PROGRAMMING LAB - II

Objective: *The lab aims to develop an understanding of different applications OOPs, GUI and Socket Programming. Web automation using Python.*

Credits:02 **L-T-P-J:0-0-2-0**

Module No.	Content	Teaching Hours
I & II	**Module 1 Lab Tasks:** Class Creation and Object Instantiation Task Description: Create a Python class 'Employee' with attributes like name, age, and salary. Instantiate multiple objects of this class and perform operations like updating salary and displaying employee details. Learning Objective: Understand class creation, object instantiation, and attribute manipulation in Python. Inheritance and Method Overriding Task Description: Create a base class 'Animal' with methods like 'make_sound'. Create subclasses like 'Dog' and 'Cat' inheriting from the 'Animal' class. Override the 'make_sound' method in subclasses to produce specific sounds for each animal type. Learning Objective: Implement inheritance and method overriding to showcase polymorphic behavior. Encapsulation and Abstraction Implementation Task Description: Create a class 'BankAccount' with private attributes like balance. Implement methods for deposit, withdrawal, and balance inquiry while encapsulating the balance attribute. Create an abstract class 'Shape' with an abstract method 'calculate_area' and implement it in subclasses like 'Rectangle' and 'Circle'. Learning Objective: Implement encapsulation to hide data and use abstraction via abstract classes and methods. **Module 2 Lab Tasks:** Multithreading Implementation Task Description: Create a Python program to simulate a simple scenario involving multiple threads performing tasks concurrently. For example, simulate a coffee shop where baristas and customers operate as threads. Learning Objective: Implement and manage multithreading concepts in Python for concurrent execution. Data Visualization and Analysis Task Description: Use Pandas and Matplotlib to analyze a given dataset (e.g., sales data, weather data) and create visualizations (bar charts, line plots, histograms) to represent trends or correlations present in the data. Learning Objective: Apply data visualization and manipulation libraries for analysis and representation of data.	48

	Socket Programming and Client-Server Communication Task Description: Create a client-server architecture using Python sockets. Establish communication between the server and multiple clients, allowing data exchange (e.g., sending messages, files) between connected clients. Learning Objective: Understand socket programming fundamentals and implement client-server communication in Python. Web Automation with Selenium Task Description: Write Python scripts using Selenium to automate web interactions. For instance, automate a login process on a web application, navigate through multiple pages, and extract information from elements. Learning Objective: Use Selenium to locate and interact with web elements for automation purposes.	

Text Books:

- Irv Kalb: Object Oriented Python "O'Reilly".
- Dusty Phillips: Python 3 Object Oriented Programming.

References Books:

- Python GUI Programming with Tkinter: Develop Responsive and Powerful GUI Applications with Tkinter

Outcome: 1. Comprehensive Understanding: Gain a comprehensive understanding of Object-Oriented Programming and advanced Python concepts.
2. Practical Application: Apply learned concepts to solve real-world problems through hands-on projects and tasks.

3. Proficiency in Python: Attain proficiency in using Python for various tasks including GUI development, data visualization, concurrency, networking, and web automation.

4. Evaluation and Assessment: Evaluate understanding through assessments, projects, and review sessions.

5. These outcomes aim to equip students with a solid foundation in Object-Oriented Programming principles and advanced Python concepts, enabling them to develop practical skills and apply Python for various tasks in software development and automation.