OBJECT ORIENTED PROGRAMMING CONCEPTS

---

1. Explain the concepts and characteristics of Object-Oriented Programming (OOP) and analyze the key differences between OOP and Procedural Programming.

2. Apply the principles of Encapsulation, Abstraction, Polymorphism, Classes, Messages Association, and Interfaces in designing and implementing OOP concepts.

3. Demonstrate the understanding of Constructor, Destructor, Copy Constructor, Structures, Class Objects, Memory Management, and Static Class Data in OOP.

4. Implement Friend Functions, Friend Classes, and the 'this' pointer to enhance the functionality and security of OOP programs.

5. Develop solutions using Function Overloading to create multiple functions with the same name but different parameters or return types.

6. Utilize Overloading Unary and Binary Operators, as well as data conversion techniques, to provide flexibility and efficiency in OOP programs.

7. Analyze the concept of Inheritance, differentiate between Base and Derived classes, explore various types of inheritance, and understand the importance of Aggregation in OOP.

8. Implement Dynamic Polymorphism to allow objects to be treated as instances of their parent class, enabling flexibility and extensibility in OOP programs.

9. Design and implement Function Templates and Class Templates to create reusable code that can work with different data types in OOP.

10. Apply Exception Handling techniques to detect and handle errors and exceptions in OOP programs, ensuring robustness and reliability in software development.