Theory

What is OOP?

-> Object-Oriented Programming (OOP) is a programming based on the concept of object

which are instances of classes. Objects can contain both data and methods that operate on the

data.OOP allows for the modeling of real-world entities and relationships, promoting code reusability,

scalability, and organization.

->Classes and Objects:

Classes: Blueprints for creating objects, defining properties (data) and behaviors (methods).

Objects: Instances of classes, representing entities with specific states and behaviors.blue print of class.

->Encapsulation:

The practice of bundling data and methods that operate on the data within a single unit (class).

->Inheritance:

A mechanism for creating new classes based on existing ones. It allows the new class (subclass)

to inherit properties and behaviors from the existing class (superclass), promoting code reuse and hierarchy.

->Polymorphism:

The ability of different classes to be used again through a common interface.

It allows for methods to be defined in the superclass and overridden in subclasses.

-two type

-method overload,method overriding

->Abstraction:

The process of hiding complex implementation details and showing only the necessary features of an object.

It simplifies interaction with objects.

->Difference between OOP and POP

->OOP (Object-Oriented Programming):

Structure: Organizes code into classes and objects, focusing on data and behaviors encapsulated within objects.

Modularity: Promotes code reuse through inheritance and polymorphism, making it easier to extend and maintain code.

Data Security: Provides encapsulation, allowing data hiding and protection.

Complexity: Well-suited for complex and large software systems, modeling real-world entities and relationships.

->POP (Procedural-Oriented Programming):

Structure: Organizes code into procedures or functions, focusing on sequential execution of instructions.

Modularity: Code reuse is achieved through functions and procedures, but lacks inheritance and polymorphism.

Data Security: Data is often shared globally, leading to potential issues with data integrity and security.

Complexity: More suited for simple and smaller programs with straightforward, linear tasks.