Object-Oriented Programming

An Overview of Key Concepts and Practices

What is OOP?

Object-Oriented Programming (OOP) is a type of programming paradigm based on the concept of objects, which contain data in the form of fields (attributes) and code in the form of procedures (methods).

Key Principles of OOP

Encapsulation

- Restricts direct access to data members

-Uses access modifiers private, public, protected etc

-Improves security and data integrity

Inheritance

-Allows a class to inherit properties and methods from another class

-Promotes code reusability

Polymorphism

-Allows a single method to perform different tasks

-Improves flexibility and scalability of the code

Abstraction

Polymorphism

-Hides complex implementation details

-Provides only essential features to the user

-Achieved through abstract classes and interfaces

Definition of an Object

Object

An object is an instance of a class. It represents a real-world entity or concept in a program. It contains data (attributes) and methods (behaviors) that operate on the data

Example:

Object: cat1
Class: Cat
Attributes: color, breed, age
Methods: meow(), eat()

Definition of a Class and its make-up

Class

A class is a blueprint or template for creating objects. It defines the attributes and methods that the objects of this class will have.

Class Structure

Attributes (Fields): Variables that hold the state of an object.

Methods: Functions that define the behavior of objects.

Constructor: A special method used to initialize an object when it's created.

Class and Object Association

Relationship

- A class is like a blueprint and a object is a specific instance of that class.
- You can create multiple objects from a single class.
- An object is created using the *new* keyword followed by the constructor of the class.

OOP Development Process

Identify the problem
Understand the requirements of the system and identify the objects that will help solve the problem

Design Classes
Define the classes, attributes and methods

Implement the Classes
Code the classes and their behaviors

Create Objects
Instantiate objects from the defined classes and begin using them

Test the System
Test individual objects and interactions to ensure everything behaves as expected