Part -6

What is paradgim and it types ?

What is oop ?

Why oop’s is introduced in python ?

Key Principles of OOP in Python ?

What is the use of oop’s in the real world ?

Advantages of OOP’s

Disadvantages if OOP

What is paradigm ?

A **"paradigm"** refers to a distinct set of concepts, patterns, practices, or models within a particular field, guiding the way something is understood, approached, or accomplished

In the context of computer science and programming, a paradigm represents a **fundamental style or way of thinking about and structuring code.**

Types of paradigms

- Imperative Programming Paradigm
- Declarative Programming Paradigm
- Object-Oriented Programming (OOP) Paradigm 
- Functional Programming Paradigm
- Procedural Programming Paradigm
- Logic Programming Paradigm

techcodio

What is OOP?

- It is stand's for **Object-Oriented Programming (OOP)**
- IT is a programming paradigm that revolves around the concept of objects, which can contain data (attributes) and code (methods/functions).
- Attributes in Python refer to the characteristics or properties associated with an object

Why oop's is introduced in python ?

- Object-Oriented Programming (OOP) was introduced in Python to provide developers with a powerful and organized way to write code. Python implemented OOP principles to enhance code **reusability, modularity, and maintainability**

techcodio

Key Principles of OOPs in Python ?

Classes

Objects

Abstraction

Inheritance

Encapsulation

Polymorphism

techcodio

What is the use of oop’s in the real world ? Applications

- Software Development:
- Graphical User Interface (GUI) Development
- Game Development
- Web Development
- Database Management
- Simulation and Modeling
- Embedded Systems and IoT
- Financial and Business Applications
- AI and Machine Learning

Advantages of OOP’s

1. **Modularity:** Classes and objects facilitate code organization, making it easier to maintain, understand, and reuse code.
2. **Code Reusability:** Inheritance allows the creation of new classes with shared attributes and methods from existing ones, reducing redundancy.
3. **Encapsulation and Security**: Access to certain attributes and methods can be restricted, enhancing security and preventing unintended modifications.
4. **Flexibility and Scalability**: OOP promotes a flexible and scalable design, allowing the addition of new features without affecting existing code.

Disadvantages of OOP’s

- Complexity
- Overhead
- Memory Consumption
- Code Readability
- Not Always the Best Approach