1. What is OOP? List OOP concepts.

🡪OOP stands for Object-Oriented Programming, which is a programming paradigm that uses objects and their interactions to design applications. The basic idea of OOP is to break down complex problems into smaller, more manageable objects, and to organize these objects into classes that encapsulate data and behavior.

Here are some of the key concepts of OOP:

Class: A blueprint or template for creating objects. It defines the data and behavior of the objects.

Object: An instance of a class. It has its own state and behavior, but is created using the definitions and methods of the class.

Encapsulation: The concept of hiding the implementation details of a class from other objects. It allows for data to be protected and accessed only through specified methods.

Inheritance: The process of creating a new class by inheriting properties and behaviors from an existing class. It allows for the reuse of code and the creation of hierarchies of classes.

Polymorphism: The ability of objects of different classes to respond to the same message or method in different ways. It allows for flexibility and extensibility in an application.

Abstraction: The process of reducing complex objects into simpler and more manageable ones. It allows for the creation of interfaces and abstract classes that define common behaviors without implementing them.

Overall, OOP provides a powerful way to organize and structure code, making it more modular, reusable, and easier to maintain.

2. What is the difference between OOP and POP?

🡪OOP (Object-Oriented Programming) is a programming paradigm that is based on the concept of objects. In OOP, software is organized around objects, which are instances of classes that encapsulate data and behavior. Objects interact with each other by sending messages and calling methods. OOP allows for code reusability, modularity, and abstraction, making it easier to maintain and extend large codebases.

POP (Procedural Oriented Programming) is a programming paradigm that is based on the concept of procedures. In POP, software is organized around procedures, which are sets of instructions that perform a specific task. Procedures can be organized into functions or modules to facilitate code reuse. In POP, data and behavior are separated, and the focus is on the sequence of operations rather than the objects that perform them.

The key difference between OOP and POP is that OOP is based on the concept of objects, while POP is based on procedures. OOP allows for code reusability, modularity, and abstraction, while POP focuses on the sequence of operations and data processing. OOP is well-suited for complex systems that require modularization and extensibility, while POP is simpler and more suited for small programs or scripts.