Object-Oriented Programming (OOP):-

Concepts Object-oriented programming (OOP) is a paradigm that enhances code organization by centering it around objects—units that encapsulate both data and methods. OOP introduces fundamental principles contributing to modular, reusable, and scalable software development:Encapsulation:Definition: Encapsulation involves bundling data (attributes) and methods (functions) into a cohesive unit called a class. Purpose: Conceals an object's internal state, allowing interactions only through designated methods, enhancing security and control.Abstraction:Definition: Abstraction simplifies complex systems by modeling classes based on shared essential properties and behaviors. Purpose: Focuses on what an object does rather than the intricacies of its implementation, enhancing clarity and reducing complexity.Inheritance:Definition: Inheritance enables a class to inherit properties and methods from another class, establishing a hierarchical parent-child relationship. Purpose: Promotes code reuse, efficiency, and the creation of a structured class hierarchy.Polymorphism:Definition: Polymorphism enables objects to be treated as instances of their parent class, providing a unified interface for different types. Purpose: Facilitates flexibility in handling diverse data types or objects, enhancing adaptability and simplifying code. These principles collectively contribute to the development of maintainable, scalable, and efficient software systems. OOP encourages modular, understandable, and reusable code, fostering an intuitive and organized approach to software development.