Object-Oriented Programming (OOP) is a programming paradigm that uses "objects" to represent data and methods to manipulate that data. It is based on several key concepts:

1. **Classes and Objects**:
   * A **class** is a blueprint for creating objects. It defines a set of attributes (data) and methods (functions) that the created objects will have.
   * An **object** is an instance of a class. It contains actual values and can perform actions defined by its class.
2. **Encapsulation**:
   * This principle involves bundling the data (attributes) and methods (functions) that operate on the data into a single unit or class. It restricts direct access to some of the object's components, which can prevent the accidental modification of data. Access to the data is typically controlled through public methods (getters and setters).
3. **Inheritance**:
   * Inheritance allows a new class (subclass or derived class) to inherit attributes and methods from an existing class (superclass or base class). This promotes code reusability and establishes a hierarchical relationship between classes.
4. **Polymorphism**:
   * Polymorphism allows methods to do different things based on the object it is acting upon, even if they share the same name. This can be achieved through method overriding (in subclasses) and method overloading (same method name with different parameters).
5. **Abstraction**:
   * Abstraction is the concept of hiding the complex implementation details and showing only the essential features of the object. This simplifies the interaction with the object and reduces complexity.

OOP is widely used in software development because it helps in organizing code, making it more modular, reusable, and easier to maintain. Popular programming languages that support OOP include Java, C++, Python, and C#.