1. Class:

- Represents a blueprint or template for creating objects.

- Advantages: Encapsulates data and behavior, promotes code reusability, supports inheritance for code organization.

- Disadvantages: Can introduce complexity, may require more memory and processing power.

2. Attribute:

- Represents a property or characteristic of an object.

- Advantages: Stores information specific to an object, facilitates data management within classes.

- Disadvantages: Overuse can lead to cluttered classes, improper use may violate encapsulation.

3. Method:

- Defines behavior or actions that objects of a class can perform.

- Advantages: Encapsulates related functionality, promotes code organization and readability.

- Disadvantages: Large number of methods can make classes unwieldy, may introduce dependencies between classes.

4. Object:

- Represents an instance of a class, embodying the properties and behaviors defined by the class.

- Advantages: Encapsulates data and behavior, supports interaction between components, facilitates modular design.

- Disadvantages: Creation and management of many objects can impact performance, improper use can lead to code duplication.