OOPS

**Basic OOP Concepts in C#**

1. **What are the basic principles of OOP?**
   * **Encapsulation**
   * **Abstraction**
   * **Inheritance**
   * **Polymorphism**
2. **What is the difference between a class and an object?**
   * **Class**: A blueprint for creating objects.
   * **Object**: An instance of a class.
3. **What is encapsulation in C#?**
   * Wrapping data and code into a single unit (class), hiding internal state using access modifiers (private, public, etc.).
4. **What is abstraction?**
   * Hiding complex implementation details and exposing only essential features.
5. **What is inheritance in C#?**
   * One class (child) inherits members of another class (parent) using the : symbol.
6. **What is polymorphism?**
   * One interface, many implementations. Achieved through method overloading and overriding.

**Advanced Concepts**

1. **What is method overloading vs. method overriding?**
   * **Overloading**: Same method name, different signatures in the same class.
   * **Overriding**: Derived class provides a specific implementation using override.
2. **What is the use of virtual, override, and new keywords in C#?**
   * virtual: Marks a method that can be overridden.
   * override: Used in derived class to override base method.
   * new: Hides the base class member (not override).
3. **What is an abstract class?**
   * A class that cannot be instantiated. May contain abstract methods (without implementation).
4. **What is an interface in C#?**
   * A contract that classes must follow. Cannot contain implementation, only declarations.
5. **Can a class inherit from multiple classes in C#?**
   * No, C# does not support multiple inheritance with classes (only one base class). But it allows multiple interfaces.
6. **What is the difference between an interface and an abstract class?**

| **Feature** | **Abstract Class** | **Interface** |
| --- | --- | --- |
| Inheritance | Single | Multiple |
| Members | Fields + Methods | Only Methods/Properties |
| Implementation | Can have it | Cannot (prior to C# 8.0) |

1. **What is the difference between composition and inheritance?**
   * **Composition**: Has-a relationship (more flexible).
   * **Inheritance**: Is-a relationship.
2. **What is a constructor and its types?**
   * A special method to initialize objects.
   * Types: Default, Parameterized, Static, Copy Constructor.
3. **What is the use of the base keyword in C#?**
   * Used to access base class members from a derived class.

### ****Practical/Behavioral Questions****

1. **Explain how you’ve used OOP in your past C# projects.**
2. **Have you used interfaces to decouple components? How?**
3. **What’s the difference between tight coupling and loose coupling?**
4. **How do SOLID principles relate to OOP in C#?**
5. **Can you give an example of real-world inheritance and polymorphism in a C# application?**