**What types of problems Design pattern solves**

Design patterns solve a variety of problems in software development. Here are some common types of problems that design patterns address:

1. Object Creation: Design patterns help with creating objects in a flexible and reusable manner. Creational patterns such as Factory Method, Abstract Factory, and Builder provide ways to create objects without specifying their exact classes or constructor parameters.
2. Object Interaction: Patterns help define how objects interact with each other, promoting loose coupling and flexibility. Behavioral patterns like Observer, Mediator, and Command enable communication and coordination between objects in a more decoupled way.
3. Object Structure: Structural patterns deal with the composition of classes and objects, providing ways to form larger structures from individual components. Examples include Composite, Decorator, and Adapter patterns, which allow for flexible object composition and functionality extension.
4. Object Lifecycle: Patterns address the management and lifecycle of objects. For example, the Singleton pattern ensures that only a single instance of a class is created, while the State pattern allows an object to change its behavior dynamically based on its internal state.
5. Performance and Optimization: Some patterns focus on optimizing system performance or improving resource utilization. Examples include Flyweight, which allows sharing of common data among multiple objects to save memory, and Proxy, which provides a lightweight substitute for a resource-intensive object.
6. Code Organization and Reusability: Design patterns promote modular and reusable code structures. They help in organizing code into smaller, more manageable units and provide guidelines for structuring classes and their relationships.
7. Scalability and Extensibility: Patterns assist in building scalable and extensible systems. Patterns like Strategy, Template Method, and Bridge promote flexibility by allowing components to be replaced or extended easily without affecting the rest of the system.
8. Error Handling and Exception Handling: Some patterns address error handling and exception propagation. The Chain of Responsibility pattern allows a request to be passed through a series of handlers until one of them can handle it, providing a flexible way to handle exceptions.