1.**Single Responsibility Principle (SRP)- definition (according to SRP each function or class should only do one thing.)** but this notification class is trying to do too many jobs at once. It does emails, SMS and telegram notifications all by itself.

2.**Open/Closed Principle (OCP)-** **definition (each function should be able to add new functionality without changing existing code.)** Here the violation is to add a new notification type, we must open the existing notification class and modify its send () method. This notification code is not open for extension, but open for modification.so every time when we touch the method, we risk breaking existing working channels.

3.**Liskov Substitution Principle (LSP)-** **definition (Objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program).**in this code subclasses can’t correctly substitute the parent class because the parent force them to handle irrelevant data. or in other word we can’t easily swap notification types because they don’t share a common interface.

4.**Interface Segregation Principle (ISP)-** **definition (Clients should not be forced to depend on interfaces they do not use. Clients are forced to depend on parameters they don’t need.)** Here our email notification shouldn’t care about phone numbers.

5.**Dependency Inversion Principle (DIP)-High-level modules should not depend on low-level modules**. here the notification code depends directly on specific notification types instead of abstract interfaces.