**Open/Closed Prensibi (Açık/Kapalı Prensibi)**, SOLID prensiplerinden biridir ve şu şekilde ifade edilir:

“Bir yazılım varlığı (class, modül, fonksiyon vb.) **genişletilmeye açık**, ancak **değiştirilmeye kapalı** olmalıdır.”

**✅ İyi Yönü:**

**Esneklik ve Bakım Kolaylığı**

• Yeni özellikler eklemek için mevcut kodu değiştirmek yerine genişletebiliriz.

• Bu, eski kodun bozulma riskini azaltır ve bakım maliyetlerini düşürür.

**Örnek:** Bir ödeme sistemi düşünelim. Yeni bir ödeme yöntemi (kredi kartı, PayPal, kripto vb.) eklemek istediğimizde, mevcut “Ödeme” sınıfını değiştirmek yerine yeni bir sınıf ekleyerek genişletebiliriz.

**❌ Kötü Yönü:**

**Aşırı Soyutlama ve Karmaşıklık**

• Kodun genişletilebilir olması için fazla soyutlama yapmak gerekebilir.

• Eğer gereksiz yere uygulanırsa, kod okunurluğu ve yönetimi zorlaşabilir.

**Örnek:** Basit bir hesaplama yapmak için tek bir fonksiyon yeterliyken, her şeyi genişletilebilir yapmak adına karmaşık bir sınıf hiyerarşisi kurmak gereksiz olur. Bu da **over-engineering** (aşırı mühendislik) problemine yol açabilir.

Özetle, Open/Closed prensibi **doğru uygulandığında** kodunuzu esnek ve sürdürülebilir hale getirir, ancak gereksiz yere kullanılırsa kodu gereksiz yere karmaşıklaştırabilir.