**Inveon Bootcamp  
 1. Hafta Ödev**

**Bölüm 1: SOLID Prensipleri**

**1. Single Responsibility Principle (SRP)**

* **Görev:**  
  Bir sınıfın birden fazla sorumluluğu olduğunda nasıl sorun yaratabileceğini kodlayın. Daha sonra bu yapıyı SRP’ye uygun şekilde yeniden düzenleyin.

**2. Open/Closed Principle (OCP)**

* **Görev:**  
  Bir sınıfa yeni bir özellik eklemek için mevcut kodu değiştirmek zorunda kaldığınız bir yapı oluşturun. Ardından bu yapıyı genişletilebilir ancak değiştirilemez hale getirin.

**3. Liskov Substitution Principle (LSP)**

* **Görev:**  
  Alt sınıfların, üst sınıfın yerine geçemeyeceği bir yapı oluşturun. Daha sonra bu yapıyı LSP’ye uygun hale getirin.

**4. Interface Segregation Principle (ISP)**

* **Görev:**  
  Fazla büyük bir arayüz oluşturun ve bu arayüzü uygulamak zorunda kalan sınıfların gereksiz metotları doldurduğu bir yapı oluşturun. Ardından bu yapıyı ISP’ye uygun hale getirin.

**5. Dependency Inversion Principle (DIP)**

* **Görev:**  
  Yüksek seviyeli bir modülün düşük seviyeli bir modüle doğrudan bağımlı olduğu bir yapı kodlayın. Ardından bu bağımlılığı tersine çevirerek DIP’ye uygun bir yapı oluşturun.

**Bölüm 2: Asenkron Programlama**

**1. Asenkron Metot Yazımı**

* **Görev:**  
  Uzun süre alan bir işlemi eşzamanlı ve asenkron olarak gerçekleştiren iki farklı metot yazın.
* Task sınıfının tüm static methodları örnek senaryolar ile açıklayın.

**3. Async/Await Kullanımı**

* **Görev:**  
  Asenkron bir işlemi async/await yapısıyla kodlayın. İşlem sırasında hata yönetimini de ekleyin.

**Bölüm 3: API Geliştirme En İyi Uygulamaları**

**1. API Endpoint İsimlendirme**

* **Görev:**  
  Bir kütüphane yönetim sistemi için RESTful API endpoint’lerini doğru şekilde isimlendirin. CRUD işlemleri için her endpoint’i kodlayın.

**2. Sayfalama (Pagination)**

* **Görev:**  
  Büyük bir veri kümesi için bir API endpoint’inde sayfalama yapısını oluşturun. İstemciden gelen sayfa numarası ve sayfa boyutuna göre sonuç döndüren bir yapı yazın.

**3. Hata Yönetimi**

* **Görev:**  
  API’de bir hata yönetimi sistemi kodlayın. Farklı hata türleri için özel mesajlar dönen bir yapı oluşturun.

**4. Önbellekleme (Caching)**

* **Görev:**  
  Bir API cevabını Redis kullanarak önbelleğe alıp, istemci aynı isteği tekrar yaptığında veriyi önbellekten dönen bir yapı oluşturun.

**Not:** Ödevlerinizi kendi GitHub hesabınızda bir repository oluşturarak paylaşın. Her hafta için ayrı bir branch oluşturup ödevlerinizi o branch üzerinden yapın. Ödev teslim süresi 7 gündür.