

**Son Teslim Tarihi: 8 Ocak 2020 Cuma 17.00**

**Konu:** Bu projede ilişkisel veri tabanlarının mantığına dayanan, küçük ölçekli ve dosya işlemleri tabanlı ders kayıt uygulaması gerçekleştirilecektir.

**İşlevsel Gereksinimler:**

- 1- Ders kayıt sisteminde öğrencinin açılan derslere kaydolması, kayıt silmesi ve kayıt güncelleme işlemlerini yapabilmesi gerekmektedir.
- 2- Açılan dersler dosyasında her bir dersin benzersiz olan ders kodu, adı, toplam kredisi, kontenjanı ve dersi veren öğretim üyesinin ID numarası bulunmalıdır. Bu dosyaya yeni ders ekleme, silme ve güncelleme işlemleri yapılabilmelidir. Bir dersi sadece bir öğretim üyesinin verdiği varsayılacaktır.
- 3- Öğretim üyeleri dosyasında her bir öğretim üyesinin benzersiz olan ID'si, Adı, Soyadı ve unvan bilgileri olmalıdır. Bu dosyaya yeni öğretim üyesi ekleme, silme ve güncelleme işlemleri yapılabilmelidir.
- 4- Öğrenciler dosyasında, her bir öğrencinin benzersiz olan öğrenci numarası, adı, soyadı, kaydolduğu toplam ders sayısı ve kaydolduğu toplam kredi bilgisi bulunmalıdır.
- 5- Bir öğrenci çok sayıda ders alabilirken, bir dersi çok sayıda öğrenci de alabileceği için ÖğrenciDersKayıt isimli bir dosyada hangi öğrencinin hangi dersi aldığı <öğrenciNO-DersKodu> eşleştirmesi yöntemiyle saklanmalıdır. Her bir eşleştirme için 1'den başlayarak otomatik artan benzersiz ID olmalıdır. Ayrıca bu eşleştirmede öğrencinin derse kayıt durumu (KAYITLI ya da BIRAKTI), derse kaydolma tarihi de saklanmalıdır.
- 6- Öğrenciler derse kaydolarken kullanıcıdan alınan maksimum ders sayısı ve maksimum kredi değerine göre ders ekleme yapılabilmelidir. Bu değerlere ulaşıncaya kadar ders ekleme işlemi engellenmelidir.

**Kodlama Gereklilikleri:**

Programlarınızda static ve global değişken kullanımı yasaktır.

Her türlü bellek tahsis dinamik bellek yönetimi fonksiyonları ile yapılmalıdır.

Yukarıda detayları verilen her bir dosyadaki nesne için uygun structure tanımlamalarını yapınız.

Oluşturulacak dosyalar text dosya formatında olmalıdır.

Aşağıdaki işlemleri yapacak bir menü ve her bir işlem için fonksiyon tasarlayınız. Benzer mantıkla çalışan fonksiyonlarda tekrardan kaçınmak için fonksiyon pointer kullanınız.

- 1- Her bir dosya için yapılacak ekle, sil ve güncelleme işlemlerini fonksiyonlarda yapınız.
- 2- Öğrenci, Ders ve öğretim üyesi için ders ekleme, silme ve güncelleme işlemlerini yapınız.
- 3- ID'si verilen öğretim üyesinin verdiği dersleri listeleyen fonksiyonu yazınız.
- 4- Ders kodu verilen bir derse kayıtlı olan tüm öğrencileri listeleyen fonksiyonu yazınız.
- 5- Öğrenci numarası verilen bir öğrencinin aldığı tüm dersleri listeleyen fonksiyonu yazınız.
- 6- ID'si verilen öğretim üyesinin verdiği bir derse ait kod kullanarak dersi alan tüm öğrencilerin sınıf listesini DERSKODU\_SINIFLISTESİ.txt dosyasına yazan fonksiyonu yazınız.

**Projenin Raporlanması:**

Raporunuzda, yazdığınız her bir fonksiyonu test etmek için yeterli sayıda öğrenci, ders ve öğretim üyesi bilgisi girerek yazdığınız fonksiyonların doğru çalıştığını gösteriniz. Bu amaçla programı çalıştırdığınızda ekrana çıkan ve kullanıcının ekran üzerinden giriş yaptığı her şeyi komut satırından kopyalayıp raporunuzu oluşturacak Word dosyasına ekleyiniz. Program sona erdiğinde verilerin saklandığı tüm dosyaların içeriklerini de raporunuza ekleyiniz.

**TESLİM EDİLECEKLER:**

**ÖĞRENCİNO.RAR** Dosyası içinde **ÖğrenciNo.docx** rapor dosyası, **ÖğrenciNo.c** kaynak kodu ve test koşumu sonucunda ortaya çıkan dosyalar olmalıdır. Benzer ödevler kopya olarak değerlendirilecek ve ilgili her ödev 0 olarak değerlendirilecektir.

Başarılar Dileriz.