

# ÖDEV 10

Öğrenci ve Ders adında iki yapının yanı sıra, aşağıda açıklanmıştır:

Öğrenci Alanları: isim, soyadı, öğrenci kimliği, mid1Grade, mid2Grade, finalGrade, ortalama

Dersin Alanları: courseName, courseCode, myStudentArray (Öğrenci yapıları dizisi), currentStudentCount

Fonksiyonlar:

void createNewStudent (struct Course \* myCourse);

void setGradeOfStudent (struct Course \* myCourse);

void findAndDisplayAverage (struct Course \* myCourse);

struct Student \* findStudentByID (int id, struct Kursu \* myCourse);

void displayAverageOfAllStudents (struct Course \* myCourse);

void displayAverageOfStudentsInInterval (struct Course \* myCourse);

Ana işlevde, kurs adı ve kursu için tek bir Kurs yapısı başlatacaksınız kodu sırasıyla "Bilgisayar Programlamamın Temelleri" ve "CSE114" olur. Bir döngüde, aşağıdaki menüyü görüntüleyeceksiniz:

1. Yeni öğrenci oluşturun
2. Öğrencinin notunu belirleyin
3. Bir öğrencinin ortalamasını bulma ve görüntüleme
4. Tüm öğrencilerin ortalamasını göster
5. Öğrencileri bir aralıkla ortalama görüntüleme
0. Çıkış

Menüdeki her öğe için, prototipi verilen ilgili yöntemi çağırırız.

Bu işlevlerin tanımları aşağıda açıklanmaktadır:

createNewStudent: Kullanıcıdan yeni öğrencinin adını, soyadını ve kimliğini girmesini isteyin . Kullanıcı tarafından girilen değerler, öğrencinin öğrencinin bulunduğu alanlara atanır. *myStudentArray* ders değişkenin tarafından işaret *MYCOURSE* . *currentStudentCount* güncellenecek böylece daha sonra yerleştirilen öğrenci için ayrılan alanı belirler.

setGradeOfStudent: Kullanıcıdan notları olacak öğrencinin kimliğini girmesi istenir ayarlayın. *findStudentByID* belirtilen id karşılık gelen öğrenci bulmak için çağrılır.

findAndDisplayAverage: Kullanıcıdan, öğrencinin kimliğini girmesi istenir. ortalama hesaplanacaktır. *findStudentByID* öğrenci tekebül bulmak için çağrılır belirtilen kimlik. Hesaplanan ortalama değeri, göstermeden önce öğrenci.

findStudentByID: Traversler *myStudentArray* ders değişkeninde tarafından işaret *MYCOURSE* için kimliği bu işlevin parametresi olan kimliğe uyan öğrenciyi bulun. Eğer bir Ekleştirme öğrenci, var olan *findStudentByID* aksi takdirde NULL döndürür onun işaretçi döndürür.

displayAverageOfAllStudents: Traversler *myStudentArray* sıvri ders değişken bölgesi tarafından *MYCOURSE* ekran için *isim* , *soyadı* , *kimliği* ve *ortalama* tüm öğrenciler alanlarında.

displayAverageOfStudentsInInterval: Kullanıcıdan en düşükAvg ve en yüksek Traverse *myStudentArray* ders değişkeninde tarafından işaret *MYCOURSE* için Ekran *adı* , *soyadı* , *kimliği* ve *ortalama* aralıkta ortalama olan öğrencilerin alanları [LowestAvg, highestAvg].

**UYARI:**

- biçimi sadece kaynak dosyası glünderin assignment10\_name\_surname.c
- Emin olun • dosyanızın uzantısı *.c* . Uzantıyı nasıl kontrol edeceğinizi bilmiyorsanız lütfen COADSYS'deki dosyaya ("Kodunuzun nasıl çalıştırılırınız?") bakın.
- Dışında herhangi bir kütüphane başka kullanmayın • *stdlib* ve *dize* ( *İpucu: strcpy yararlı olabilir* ).

- Dizilerinizi sabit bir boyut verin. Kullanıcı tarafından hiçbir giriş yapılmayacağını varsayalım. bu boyutu asıyor. Örneğin, için *myStudentArray* [ *5* ], varsayalım *createNewStudent* daha kullanıcı birden tarafından çağrılan olmayacak *5* kere.
- Var olmayan bir kimlik vermek gibi geçersiz testler olmayacağını varsayalım. *setGradeOfStudent* .
- Coadsys ile ilgili sorularınızı sorun.
- Bu görevden Deniz Tuana Ergonul sorumludur.

Örnekler: