

Ödev raporlarının hazırlanması ve sisteme yüklenmesi hakkında uyulacak kurallar!

OBS sınav modülü üzerinden gönderilen ödevlerin raporları tanımlanan zaman aralığında <https://classroom.google.com/c/NzI3NzA5OTE2NTU3?cjc=rnixhtz> bağlantısında bulunan, rnixhtz kodu ile girilen sadece bu ders için tanımlanmış sınıf platformuna yüklenecek, kaydedilecektir. Daha sonra OBS, email veya başka bir kanaldan gönderilmemelidir, **classroom dışında bir yere yüklenmemelidir.**

Ödevler Word dökümanı olarak aşağıdaki kurallara uygun biçimde hazırlanacak, sisteme yüklenmeden önce pdf formatına çevrilecektir.

Aşağıdaki tablo araçlarıyla oluşturulan şablon kullanılacaktır. Sayfa kenar boşlukları 1,27cm (dar) biçimde seçilmelidir, yazı font boyutu 11'den büyük olmamalıdır, ödevler sıralı olmalıdır. Ödevler tanımlanacak zaman aralığında sisteme yüklenmelidir. Daha sonra email ile veya başka bir kanaldan gönderilmemelidir.

Bu ödevde sisteme .exe veya .cpp dosyalar yüklenmemelidir. Projenizi github hesabınızdaki alana kaydedeceksiniz, kaydettiğiniz alanın indirme linkini raporun üst alanında ayrılan yere kopyalayacaksınız. Projenizi bu linkten indirerek inceleyeceğiz. **classrooma projenizin sadece raporunu yükleyeceksiniz.**

Aşağıdaki tablo şablonunu boş bir sayfaya yapıştırarak, gerektiğinde tabloya yeni satırlar ekleyerek düzgün, standart bir ödev raporu oluşturmak için itina gösteriniz. Bu ödevin Finale olan katkı yüzdesi %10 dur. (10 puandır.)

Ödevlerin programlama eğitiminize katkısı olabilmesi için öneriler;

1.aşama: Ödevleri kendinize tanıdığınız makul bir süre içerisinde kendiniz yalnız yoğunlaşarak, önce kağıt üzerinde analiz, tasarım, kaba kod, kod çalışması yapınız. Problemi yapısına uygun, mantıksal sıralı, küçük birkaç parçaya bölerek çözümleyin, kodlayın, çalıştırarak ilerleyin. Yazdığınız program tamamlanana kadar unutmayın sürekli test aşamasındadır!

2.aşama: Makul süre sonunda ödevi tamamlayamazsanız daha kısa bir süre ayırarak internet kaynaklarından takıldığınız bölümlerin çözümlerini araştırınız. Unutmayın internette aynı ödevin çözümü yoktur! İnternetteki kaynaklardan, örneklerden teknik bilgi araştırınız.

3.aşama: Gerek duyarsanız sınıf arkadaşlarınızla bilgi paylaşımına gidiniz. Bu paylaşım kesinlikle kopyalama, dosya değişimi seviyesine çıkmasın! Bu tür paylaşımlar size ve arkadaşına katkı sağlamaz! Hatayı anlatma, çözüme, kod inceleme, problemi tartışma seviyesinde kalmalıdır.

4.aşama: Katkının hiç olmadığı veya %60'dan az katkı olan çözümü ödev olarak yüklemeyin! Boş kalsın, kendini kandırma! Gönderdiğiniz ödevleri sizin yaptığınızı ve konuyu öğrendiğinizi farz ediyorum.

Başarılar diliyorum. Prof.Dr.Tuncay AYDOĞAN

ÖDEV: Dönem içerisinde öğrendiğiniz programlama konuları çerçevesinde aşağıdaki listede adınıza atanan konuda binary dosyalama içeren projeyi gerçekleştiriniz.

Projenin amacı; dosyalama, fonksiyonlar, temel veri tipleri, dizi, structure, string, döngü, kontrol, giriş, çıkış komut ve ifadeleri vb. müfredat kapsamındaki konuların olabildiğince tamamını kullanmaktır. Ayrıca; bir problemi analiz ederek ihtiyaçları belirleme, doğru çözüm yöntemlerini ortaya koymak, seçilen yöntemin algoritmasını ve kodunu geliştirmek, menü tasarlamak, kullanıcı dostu arayüz tasarlamak, projeyi raporlamaktır.

Projeye başlarken, size verilen konuya ait meslekte çalıştığınızı, yazacağınız projeyi kullanacağınızı düşünün, hangi özelliklerde olmasını beklerdiniz? Önce projenizi bir A4'e kalemle genel hat ve özellikleri ile planlayınız.

Projenin özelliklerini asgari seviyede (en çok gerekli, olmazsa olmazları dikkate alarak) tasarlayınız, ihtiyaçları belirleyiniz. **C++ dilinin string dışındaki sınıf/nesne özelliklerini henüz öğretmediğimiz için kullanmayınız.**

Sınıf/nesne tanımlamalı, pointer, listeler veriyapıları kullanılan projeler ve dosyalama özelliği olmayan projeler değerlendirmeye alınmayacaktır 0 (sıfır) puan olarak değerlendirilecektir. Raporu aşağıdaki format dışında olan, gönderdiği kaynak kodu açılmayan, dosyalama özellikleri olmayan, müfredat dışı programlama özellikleri ve diller kullananların ödevleri değerlendirmeye alınmayacaktır.

Proje; menü, dosyalanan veriler üzerinde kayıt, ara, düzelt/güncelle, sil, listele ve gerekiyorsa konuyla ilgili hesaplamaları yapabilecek fonksiyonlardan oluşmalıdır. Fonksiyonların parametre özellikleri ihtiyaçlara göre belirlenmelidir. Puanlama bu kriterler ve raporlama düzeni esas alınarak yapılacaktır.

Fikir, yöntem, hataların bulunması ve düzeltilmesi gibi konularda yardımlaşmaları mutlaka yapınız. Yardımlaşma sınırını kesinlikle aşmayınız! Zorlamayınız! Aynı projeler diğer grup öğrencilerine de verilecektir. Sınıf içerisinde kopya kod veya proje tespit edilmesi durumunda sonuçlardan ilgili öğrenciler sorumlu olacaktır. Özgünlüğü az olan projelerin değerlendirmesini yapmama kanaatim var!

Ödev teslim biçimi: Bu ödevin pdf raporu iki bölümden oluşacaktır. Birinci bölüm “projenizin bir kullanım kılavuzu” biçiminde ve aşağıdaki şablonda belirtilen özelliklerde yazılacaktır. İkinci bölümünde ise kodlarınız bulunacaktır. Ayrıca bir kapak sayfası olmayacak, ilk sayfa aşağıdaki şablon ile başlayacaktır.

OBS Sisteme pdf’ye dönüştürülmüş aşağıdaki rapor yüklenecektir.

Kaynak kodu göndereceğiniz github linkinden indireceğim denemek/çalıştırmak istediğimde herhangi bir ayar yapmama gerek kalmamalıdır. Kaynak kodun ilk satırlarına öğrenci bilgilerinizi ve proje adını açıklama olarak yazınız (/ * öğrenci ve proje bilgileri * /).

ISPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ -TEKNOLOJİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ 2024-2025 GÜZ DÖNEMİ BLG-101 ALGORİTMA VE PROGRAMLAMAYA GİRİŞ DERSİ ÖDEV RAPORU
Ad Soyad: Mustafa Yıldız Numara: 2212729001
GÜVENLİK GÖREVLİLERİ NÖBET TAKİP OTOMASYONU
Projenizin github linkini buraya kopyalayınız:
KONU: Bu proje bir çalışma ortamında güvenlik görevlileri vardiyaya gelmiş mi? Geldiyse kaçta gelmiş ve güvenlik görevlilerinin kişisel bilgilerini dosyalar
<div><pre>----- Güvenlik Nöbet Takip Sistemi ----- ->->->->-> 1- Güvenlik Görevlisi Ekle <-<-<-<-<-<- ->->->->-> 2- Görevli Listeleme <-<-<-<-<-<- ->->->->-> 3- Görevli Arama <-<-<-<-<-<- ->->->->-> 4- Görevli Silme <-<-<-<-<-<- ----- SEÇİM : </pre></div>
Güvenik Nöbet Takip Projesinin Ana Menüsü

```
C:\Users\musta\Desktop\dosy  X + v

Görevlinin Adını giriniz: Mauro
Görevlinin Soyadını giriniz: İcardi
Görevlinin Numarasını Giriniz: 9
Görevlinin Cinsiyetini Giriniz: e
Görevlisinin Nöbet Başlangıç Saatini Giriniz (HH:MM): 08:00
Görevlisinin Nöbet Bitiş Saatini Giriniz (HH:MM): 16:00

Tekrar Güvenlik Görevlisi Ekleme İster misiniz ? (e/h) : |
```

Projenin Görevli Ekleme Bölümü

```
C:\Users\musta\Desktop\dosy  X + v

|----- 1. GÖREVLİ BİLGİLERİ -----|
-> Görevlinin Adı: alper
-> Görevlinin Soyadı: serin
-> Görevlinin Numarası: 6
-> Görevlinin Cinsiyeti: e
-> Nöbet Başlangıç Saati: 18:00
-> Nöbet Bitiş Saati: 02:00

|----- 2. GÖREVLİ BİLGİLERİ -----|
-> Görevlinin Adı: Mauro
-> Görevlinin Soyadı: ~cardi
-> Görevlinin Numarası: 9
-> Görevlinin Cinsiyeti: e
-> Nöbet Başlangıç Saati: 08:00
-> Nöbet Bitiş Saati: 16:00

PROGRAMI TEKRAR ÇALIŞTIRMAK İSTER MİSİNİZ ? (e/h) : |
```

Projenin İçinde Bulunan Verileri Listeleme Bölümü

```
C:\Users\musta\Desktop\dosj  X + v

Aramak istediğiniz güvenlik görevlisinin numarasını giriniz: 6

|----- 1. GÖREVLİ BİLGİLERİ -----|
-> Görevlinin Adı: alper
-> Görevlinin Soyadı: serin
-> Görevlinin Numarası: 6
-> Görevlinin Cinsiyeti: e
-> Nöbet Başlangıç Saati: 18:00
-> Nöbet Bitiş Saati: 02:00

Tekrar arama yapmak ister misiniz? (e/h): |
```

Projenin Arama Bölümü

```
C:\Users\musta\Desktop\dosj  X + v

Silme istediğiniz güvenlik görevlisinin numarasını giriniz: 7

Görevli bulundu ve silindi:
-> Ad: Mustafa
-> Soyad: Yildiz
-> Numarası: 7
-> Cinsiyeti: e
-> Nöbet Başlangıç Saati: 00:00
-> Nöbet Bitiş Saati: 12:00

Silme işlemi başarıyla tamamlandı!

PROGRAMI TEKRAR ÇALIŞTIRMAK İSTER MİSİNİZ ? (e/h) : |
```

Projenin Silme Bölümü

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <cstdlib>
#include <conio.h>
#include <iomanip>
using namespace std;

struct guvenlik {
    char isim[20];
    char soyisim[20];
    char no[10];
    char cinsiyet;
    char baslangicSaati[10];
    char bitisSaati[10];
};

guvenlik gorevli;
```

```

void GuvenlikEkle();
void GuvenlikListeleme();
void GuvenlikArama();
void GuvenlikSilme();

int main()
{
    setlocale(LC_ALL, "Turkish");

    char secim;
    do {
        system("cls");

        cout << "|----- Güvenlik Nöbet Takip Sistemi -----|" << endl;
        cout << "|                                     |" << endl;
        cout << "|->->->->-> 1- Güvenlik Görevlisi Ekle <-<-<-<-<-|" << endl;
        cout << "|->->->->-> 2- Görevli Listeleme    <-<-<-<-<-|" << endl;
        cout << "|->->->->-> 3- Görevli Arama        <-<-<-<-<-|" << endl;
        cout << "|->->->->-> 4- Görevli Silme        <-<-<-<-<-|" << endl;
        cout << "|                                     |" << endl;
        cout << "|-----|" << endl;

        cout << "\nSEÇİM : ";
        cin >> secim;

        switch (secim) {
            case '1':
                GuvenlikEkle();
                break;
            case '2':
                GuvenlikListeleme();
                break;
            case '3':
                GuvenlikArama();
                break;
            case '4':
                GuvenlikSilme();
                break;
        }

        cout << endl;
        cout << "\nPROGRAMI TEKRAR ÇALIŞTIRMAK İSTER MİSİNİZ ? (e/h) : ";
        secim = getche();
    } while (secim == 'e' || secim == 'E');

    return 0;
}

void GuvenlikEkle()
{
    setlocale(LC_ALL, "Turkish");

    ofstream yaz("guvenlik.dat", ios::binary | ios::app);

    char secim;

    do {
        system("cls");

        cout << "\nGörevlinin Adını giriniz: ";
        cin >> gorevli.isim;

        cout << "\nGörevlinin Soyadını giriniz: ";
        cin >> gorevli.soyisim;

        cout << "\nGörevlinin Numarasını Giriniz: ";
        cin >> gorevli.no;

        cout << "\nGörevlinin Cinsiyetini Giriniz: ";

```

```

gorevli.cinsiyet = getche();
cout<<endl;
cout << "\nGörevlisinin Nöbet Başlangıç Saatini Giriniz (HH:MM): ";
cin >> gorevli.baslangicSaati;

cout << "\nGörevlisinin Nöbet Bitiş Saatini Giriniz (HH:MM): ";
cin >> gorevli.bitisSaati;

yaz.write((char*)&gorevli, sizeof(gorevli));

cout << endl;
cout << "\nTekrar Güvenlik Görevlisi Ekleme İster misiniz ? (e/h) : ";
secim = getche();
} while (secim == 'e' || secim == 'E');

yaz.close();
}

void GuvenlikListeleme()
{
    setlocale(LC_ALL, "Turkish");

    system("cls");

    ifstream oku("guvenlik.dat", ios::binary | ios::app);

    system("cls");

    oku.seekg(0, ios::end);

    int adet = oku.tellg() / sizeof(gorevli);

    if (adet > 0) {
        for (int i = 0; i < adet; i++) {
            oku.seekg(i * sizeof(gorevli)); //işaretleme
            oku.read((char*)&gorevli, sizeof(gorevli));

            cout << "\n|----- " << i + 1 << " . GÖREVLİ BİLGİLERİ -----|";

            cout << "\n -> Görevlinin Adı: " << gorevli.isim;
            cout << "\n -> Görevlinin Soyadı: " << gorevli.soyisim;
            cout << "\n -> Görevlinin Numarası: " << gorevli.no;
            cout << "\n -> Görevlinin Cinsiyeti: " << gorevli.cinsiyet;
            cout << "\n -> Nöbet Başlangıç Saati: " << gorevli.baslangicSaati;
            cout << "\n -> Nöbet Bitiş Saati: " << gorevli.bitisSaati;

            cout << endl;
        }
    } else {
        cout << "\nKAYIT BULUNAMADI" << endl;
    }

    oku.close();
}

void GuvenlikArama()
{
    setlocale(LC_ALL, "Turkish");

    system("cls");

    ifstream oku("guvenlik.dat", ios::binary | ios::in);

    oku.seekg(0, ios::end);
    int adet = oku.tellg() / sizeof(gorevli);

    if (adet <= 0) {
        cout << "\nKayıt bulunamadı!" << endl;
        oku.close();
    }
}

```

```

    return;
}

char gorevliNo[10];
char secim;

do {
    system("cls");
    cout << "\nAramak istediğiniz güvenlik görevlisinin numarasını giriniz: ";
    cin >> gorevliNo;

    oku.seekg(0, ios::beg);
    bool bulundu = false;

    for (int i = 0; i < adet; i++) {
        oku.read((char*)&gorevli, sizeof(gorevli));

        if (strcmp(gorevli.no, gorevliNo) == 0) {
            cout << "\n|----- " << i + 1 << ". GÖREVLİ BİLGİLERİ -----|";
            cout << "\n -> Görevlinin Adı: " << gorevli.isim;
            cout << "\n -> Görevlinin Soyadı: " << gorevli.soyisim;
            cout << "\n -> Görevlinin Numarası: " << gorevli.no;
            cout << "\n -> Görevlinin Cinsiyeti: " << gorevli.cinsiyet;
            cout << "\n -> Nöbet Başlangıç Saati: " << gorevli.baslangicSaati;
            cout << "\n -> Nöbet Bitiş Saati: " << gorevli.bitisSaati;
            cout << endl;

            bulundu = true;
            break;
        }
    }

    if (!bulundu) {
        cout << "\nBu numaraya ait güvenlik görevlisi bulunamadı." << endl;
    }

    cout << "\nTekrar arama yapmak ister misiniz? (e/h): ";
    cin >> secim;

} while (secim == 'e' || secim == 'E');

oku.close();
}

void GuvenlikSilme()
{
    setlocale(LC_ALL, "Turkish");

    system("cls");

    ifstream oku("guvenlik.dat", ios::binary | ios::in);
    if (!oku.is_open()) { //zorunlu
        cerr << "\nDosya açılmadı veya mevcut değil!" << endl;
        return;
    }

    ofstream yaz("gecici.dat", ios::binary | ios::out);
    if (!yaz.is_open()) { //zorunlu
        cerr << "\nGeçici dosya oluşturulamadı!" << endl;
        oku.close();
        return;
    }

    char gorevliNo[10];
    bool bulundu = false;

    cout << "\nSilmek istediğiniz güvenlik görevlisinin numarasını giriniz: ";
    cin >> gorevliNo;

```

```
oku.seekg(0, ios::beg);
while (oku.read((char*)&gorevli, sizeof(gorevli))) {
    if (strcmp(gorevli.no, gorevliNo) == 0) {
        cout << "\nGörevli bulundu ve silindi:\n";
        cout << " -> Ad: " << gorevli.isim << "\n";
        cout << " -> Soyad: " << gorevli.soyisim << "\n";
        cout << " -> Numarası: " << gorevli.no << "\n";
        cout << " -> Cinsiyeti: " << gorevli.cinsiyet << "\n";
        cout << " -> Nöbet Başlangıç Saati: " << gorevli.baslangicSaati << "\n";
        cout << " -> Nöbet Bitiş Saati: " << gorevli.bitisSaati << "\n";
        bulundu = true;
    } else {
        yaz.write((char*)&gorevli, sizeof(gorevli));
    }
}

oku.close();
yaz.close();

if (bulundu) {
    remove("guvenlik.dat");
    rename("gecici.dat", "guvenlik.dat");
    cout << "\nSilme işlemi başarıyla tamamlandı!" << endl;
} else {
    remove("gecici.dat");
    cout << "\nBelirtilen numaraya ait güvenlik görevlisi bulunamadı!" << endl;
}
}
```