



- ✓ Soru kâğıdına **adınız**, **soyadınız** ve **numaranız** dışında başka hiçbir şey yazmayınız.
- ✓ Sınav süresi **90** dakikadır.

S.1) Win32 projelerinde veya MFC uygulamalarında **olaylarla (events) iletiler (messages)** arasındaki ilişki nasıl sağlanmakta ve iletiler nasıl işlenmektedir? Gerekli gördüğünüz yerlerde **C/C++** programlama dilindeki kod parçalarını da yazarak kısaca açıklayınız. (30 p)

Puan	Harf Notu
81 - 100	AA
76 - 80	BA
70 - 75	BB
60 - 69	CB
50 - 59	CC
45 - 49	DC
40 - 44	DD
30 - 39	FD
0 - 29	FF

S.2) Bir öğrencinin herhangi bir derse ait almış olduğu 1. arasnav, 2. arasnav ve final sınavı notlarının sırasıyla % 30, % 20 ve % 50 ağırlıklarının toplamlarıyla dönem sonu ders notu hesaplanacak ve bu not yukarıda verilen tabloya göre harfli nota dönüştürülecektir. Bu işlem için bir görsel uygulama yazılmak istenmektedir. Yukarıda verilen şekildeki gibi, uygulama penceresinde verileri kullanıcıdan almak ve bunlara bağlı olarak elde edilen sonucu göstermek için **HWND** veri tipinde önceden tanımlanmış **EDIT** kontrol sınıflarıyla oluşturulmuş pencereler veya **Edit Boxes** (düzenleme kutuları) kullanıldığı ve not hesaplamasının **Hesapla** başlıklı butona tıklama olayına bağlı olarak yapıldığı varsayalım. Buna göre, programın bir

- Win32 projesi (35 p)
- MFC uygulaması (35 p)

olması durumunda, diğer işlemlerin yapıldığını varsayarak, **sadece Hesapla** başlıklı **butonun tıklama olayıyla** ilgili iletiye veya fonksiyona ait **C/C++** programlama dilindeki kod parçalarını gerekli gördüğünüz yerlerde açıklama da yaparak yazınız.



KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü
2010-2011 Eğitim-Öğretim Güz Yarıyılı
Windows Programlama 2. Arasnav Soruları



1. Herhangi bir Win32 uygulamasında bir olay meydana geldiğinde Windows olayla ilgili olarak programa bir ileti gönderir. İleti, aslında olayla ilgili bir bilgi kümesidir. Her uygulamanın bir ileti kuyruğu vardır ve iletiler uygulamanın pencere fonksiyonunda işleninceye kadar orada tutulur. MFC programlarında ise, iletileri işlemek için pencere fonksiyonu oluşturmaya gerek yoktur. İletilerin işlenmesi için ileti haritası kullanılır. Ayrıntılar için slaytlara bakınız.

2. Gerekli olan ilgili kodlar, özellikle kırmızıyla yazılmış olanlar, aşağıda verilmiştir.
a)

```
LRESULT CALLBACK MainWndProc(HWND hWnd, UINT msg, WPARAM wParam, LPARAM lParam) {
    HWND hwndCtl = (HWND) lParam;
    switch (msg) {

        case WM_COMMAND:

            switch (wParam) {
                case BN_CLICKED:
                    if (hwndCtl == hwndButtonHesapla)
                    {
                        int iSonuc = 0;
                        int iArasnav1 = 0, iArasnav2 = 0, iFinal = 0;
                        char szArasnav1[4], szArasnav2[4], szFinal[4];
                        CString sSonuc = "";

                        GetWindowText(hwndEditArasnav1, szArasnav1, strlen(szArasnav1));
                        GetWindowText(hwndEditArasnav2, szArasnav2, strlen(szArasnav2));
                        GetWindowText(hwndEditFinal, szFinal, strlen(szFinal));

                        iArasnav1 = atoi(szArasnav1);
                        iArasnav2 = atoi(szArasnav2);
                        iFinal = atoi(szFinal);
                        iSonuc = iArasnav1*0.3 + iArasnav2*0.2 + iFinal*0.5;

                        assert(iSonuc >=0 && iSonuc <= 100);

                        sSonuc = iSonuc < 30 ? "FF" :
                            iSonuc < 40 ? "FD" :
                            iSonuc < 45 ? "DD" :
                            iSonuc < 50 ? "DC" :
                            iSonuc < 60 ? "CC" :
                            iSonuc < 70 ? "CB" :
                            iSonuc < 76 ? "BB" :
                            iSonuc < 81 ? "BA" : "AA";

                        SetWindowText(hwndEditSonuc, sSonuc);
                    }

                    else if (hwndCtl == hwndButtonKapat)
                        SendMessage(hWnd, WM_CLOSE, wParam, lParam);
                    return 0;
            }
            return 0;

        case WM_DESTROY:
            PostQuitMessage(0);
            return 0;

        default:
            return DefWindowProc(hWnd, msg, wParam, lParam);
    }
    return 0;
}
```

b)

```
void CsuruDlg::OnBnClickedHesapla()
{
    int iSonuc = 0;

    UpdateData(TRUE);

    iSonuc = m_iArasnav1*0.3 + m_iArasnav2*0.2 + m_iFinal*0.5;

    assert(iSonuc >=0 && iSonuc <= 100);

    m_sSonuc = iSonuc < 30 ? "FF" :
                iSonuc < 40 ? "FD" :
                iSonuc < 45 ? "DD" :
                iSonuc < 50 ? "DC" :
                iSonuc < 60 ? "CC" :
                iSonuc < 70 ? "CB" :
                iSonuc < 76 ? "BB" :
                iSonuc < 81 ? "BA" : "AA";

    UpdateData(FALSE);
}
```