

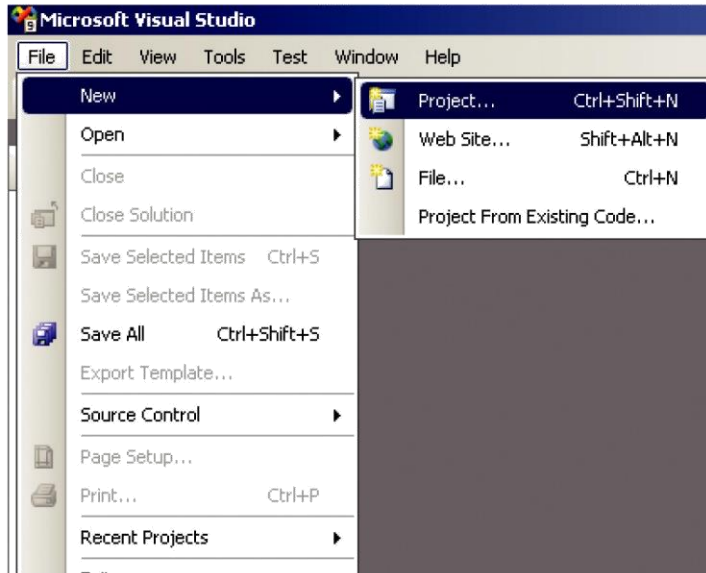
BLGM 112 – DENEY 1

MICROSOFT VISUAL STUDIO VE C PROGRAMLAMA

Amaçlar

1. Microsoft Visual Studio'nun temel özelliklerinin öğrenilmesi
2. C Programlamaya giriş
3. Microsoft Visual Studio kullanarak C programlarının çalıştırılması

Bu deneyde Microsoft Visual Studio 2008 kullanacağız, ancak bazı farklılıklar göstermelerine rağmen, diğer Tümüleşik Geliştirme Ortamlarının (Integrated Development Environment, IDE) da kullanımı benzerdir.



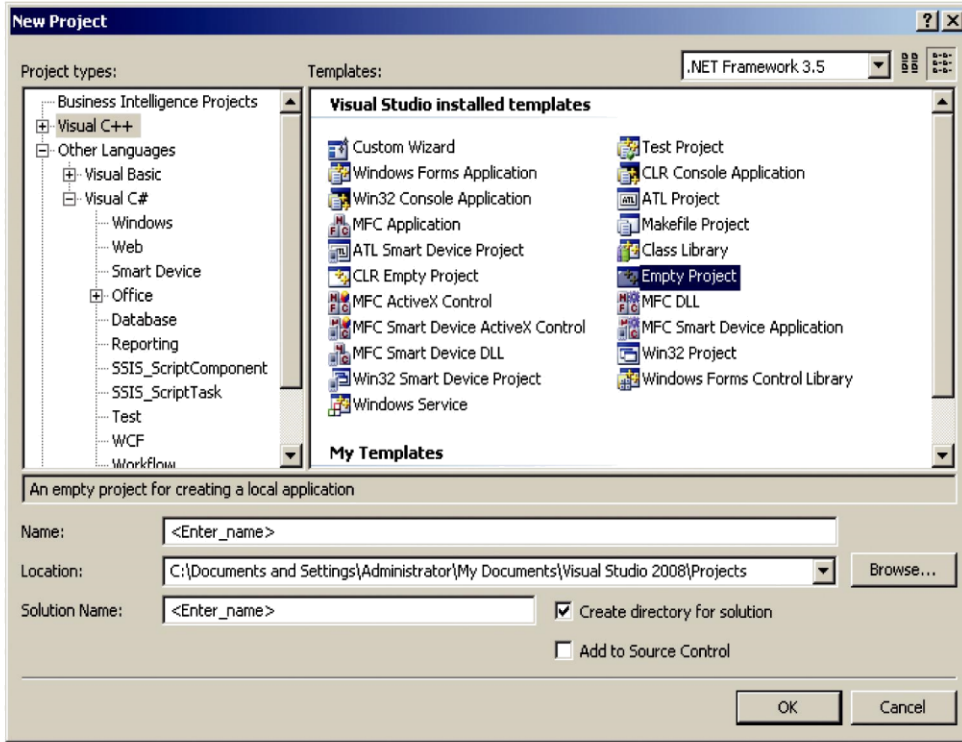
Figür 1: Dosya menüsü

“Proje tipleri” (Project types) kısmından “Visual C++” seçeneğini, yan taraftan ise “Boş proje” (Empty project) seçeneğini seçiniz. Bu adımın ardından projenize istediğiniz ismi veriniz.

Yeni Proje Oluşturma

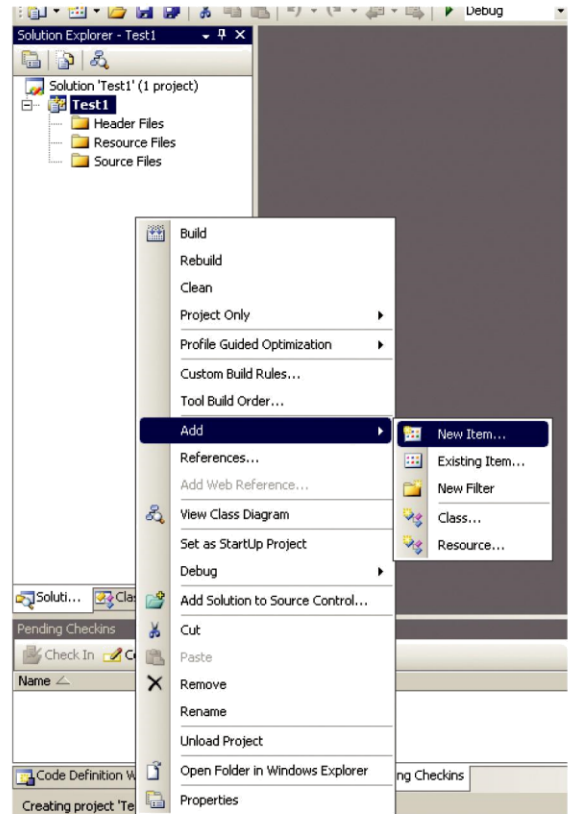
Yeni bir proje oluşturmak için öncelikle Visual Studio 2008'i Tümüleşik Geliştirme Ortamını başlatınız (Başlat → Programlar → Microsoft Visual Studio 2008 → Microsoft Visual Studio 2008). Figür 1'de gösterildiği gibi Dosya (File) menüsünden, Yeni (New) alt menüsünden Projeyi (Project) seçiniz. Bu işlem Figür 2'de gösterilen ekrana benzer bir pencere açacaktır. Bu pencerenin solunda bulunan

Projelerinize verdiğiniz isimlerin ihtiyacınız olduğunda tekrar kullanabilmeniz için akılda kalabilecek isimler olmasına özen gösteriniz. Bu adım size boş bir proje oluşturacaktır.

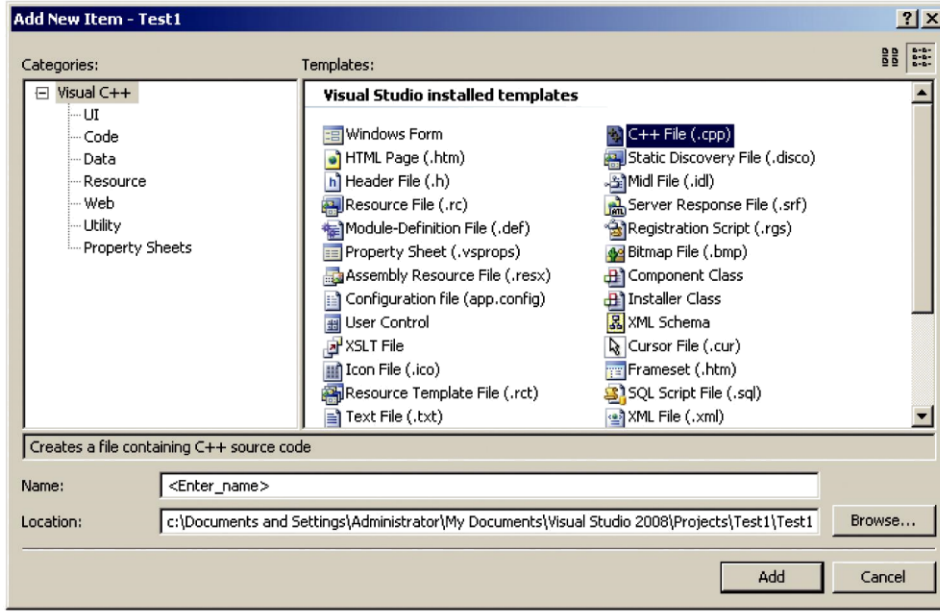


Figür 2: Yeni proje penceresi

Yeni bir projede ilk yapmak isteyeceğimiz işlem kodları yazabileceğimiz bir dosya oluşturmaktır. Bu işlemi yapabilmek için öncelikle “Proje gezgini”ni (Project explorer) bulunuz. Eğer proje gezgini penceresi görünmüyorsa “Görünüm” (View) menüsünden açabilirsiniz. Proje gezgini ekranındaki boş bir alana sağ tıklayarak Figür 3 görüldüğü gibi “Ekle” (Add) menüsünden “Yeni bileşen” (New item) seçeneğini seçiniz. Bu seçenek Figür 4'te görünen pencereyi açacaktır. Bu pencereden C++ dosyası (C++ file) seçeneğini seçerek yeni dosyanızı oluşturabilirsiniz.



Figür 3: Dosya ekleme menüsü



Figür 4: Yeni dosya oluşturma

İlk C Programı

Not: bu kısımda anlatılan örnekleri adım adım uygulamaya çalışırsanız onları anlamanız daha kolay olacaktır.

Bir C programında mevcut olması gereken ilk şey giriş fonksiyonudur. Bu fonksiyonun adının “main” olması gerekmektedir. Böylece C derleyicisi programınızı çalıştırmak için hangi fonksiyonu çağırması gerektiğini bilecektir. Bu fonksiyon geriye tamsayı (integer, int) tipinde bir değer döndürmesi gerekmektedir. Şimdilik yalnızca programın hatasız tamamlandığını belirten 0 değerini kullanacağız. Aşağıdaki örnek çalışabilecek en küçük C programıdır.

```
int main() {  
    return 0;  
}
```

Basit bir C programı olarak “Merhaba Dünya!” örneğini kullanabiliriz. Bu örnek ekrana “Merhaba Dünya!” yazısı yazarak programdan çıkar ve en temel program örneğidir. Bu işlemi yapabilmek için C'nin standart girdi/çıkı kütüphanesini eklememiz gerekmektedir. Bu işlemi aşağıdaki önışlemci bildirimini (preprocessor directive) gerçekleştirebiliriz. Bu komutun evrensel kapsamda (global scope) bulunması gerekmektedir (evrensel kapsam, fonksiyon ve diğer yapı tanımlamalarının dışında kalan alandır). Ayrıca # işaretiyle başlayan önışlemci bildirimlerinin sonuna “;” koymamanız gerekmektedir. Genel olarak önışlemci bildirimleri programınızın en üstünde yer alır. Dosyamızın en üst kısmı bu kod için ideal konumdur.

```
#include <stdio.h>
```

Bu eklentinin ardından “stdio.h” dosyasında tanımlı olan ve ekrana bilgi yazdırmak için kullanılan **printf** (print formatted, şekillendirilmiş olarak yazdır) fonksiyonunu kullanabiliriz. Aşağıdaki kod parçası konsol ekranına “Merhaba Dünya!” yazısını yazacaktır. Bu kodun main fonksiyonumuz içerisinde olması gerekmektedir. Ancak **return 0;** satırında program sonlanacağı için bu satırdan önce bulunmalıdır. Yazının sonundaki \n ibaresi bundan sonra gelecek yazıların bir sonraki satırın başından devam etmesini söyler.

```
printf("Merhaba Dünya!\n");
```

Visual Studio 2008'de çalıştırdığınız konsol uygulamaları tamamlandıklarında otomatik olarak kapanırlar. Bu yüzden bu programı çalıştırırsanız sonuç bir an ekranda görünüp kaybolacaktır. Bunu engellemek için bir MS-DOS programı olan “pause” programından yararlanacağız. Sistemdeki programları çalıştırmak için “system” fonksiyonunu kullanabiliriz, ancak, bu fonksiyon C standart kütüphanesinde bulunmaktadır. O yüzden öncelikle programımızın başına bu kütüphaneyi eklememiz gerekmektedir.

```
#include <stdlib.h>
```

Aşağıdaki komutu ise pause programını çalıştırmak için kullanabilirsiniz.

```
system("pause");
```

Bu komut çalıştığında program herhangi bir tuşa basılıncaya kadar bekler. O yüzden bu fonksiyonu hemen yazdırma fonksiyonumuzun altında çalıştırmamız gerekmektedir. Diğer programlarınızda da programınız yapacağı işlemi tamamladıktan sonra, programınızı kapatmadan önce (**return 0;** satırından önce) çalıştırarak, programlarınızın sonuçlarının ekranda kalmasını sağlayabilirsiniz.

Aşağıdaki kısımda programımızın tamamı mevcuttur.

```
#include <stdio.h>
#include <stdlib.h>

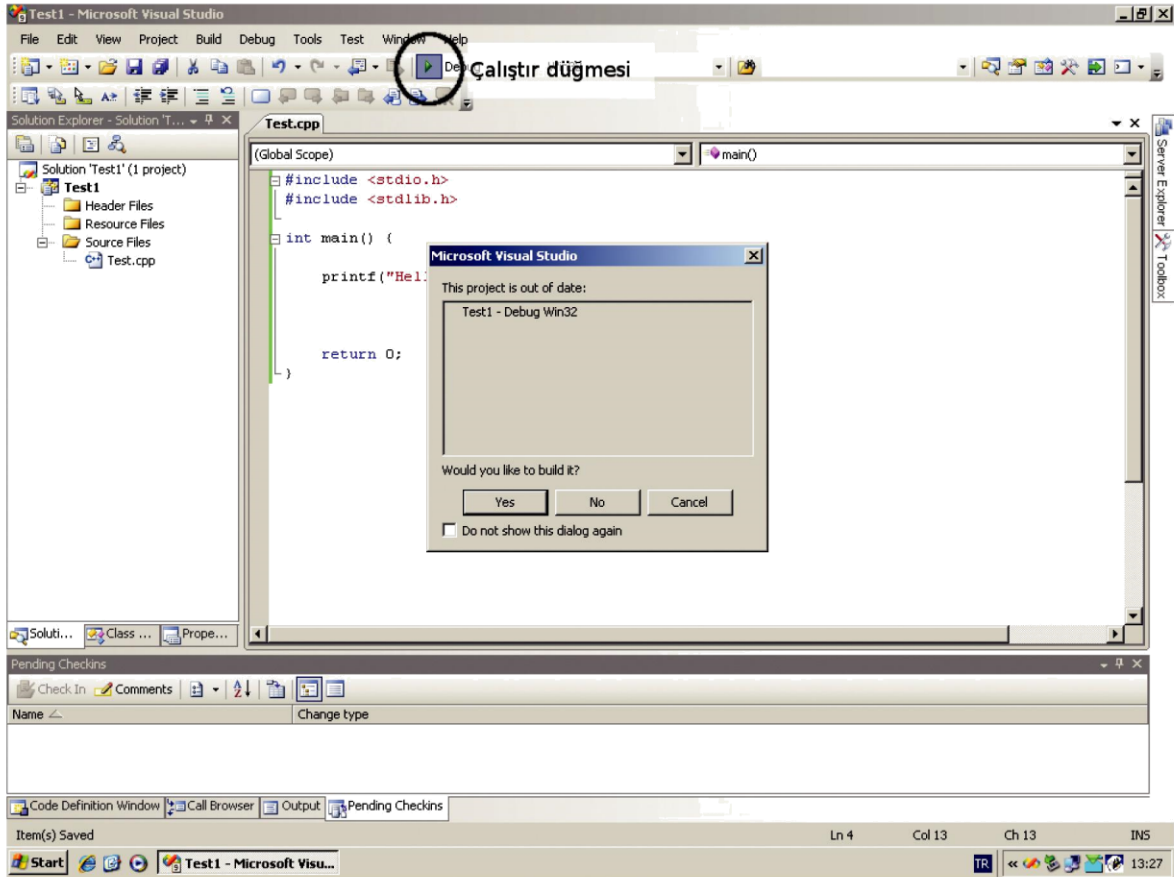
int main() {
    printf("Merhaba Dünya!\n");

    system("pause");

    return 0;
}
```

Programı Çalıştırma

Programınızı çalıştırmak için Figür 5 gösterilen çalıştır düğmesine, ya da “F5” tuşuna basabilirsiniz. Bu tuşa bastığınızda Figür 5 gösterilen ekran karşınıza gelecektir. Bu ekranda programınızı derlemek isteyip istemediğiniz sorulmaktadır. Yazdığımız kodları çalıştırabilmek için programımızı derlememiz gerektiğinden bu soruya “Evet” (Yes) cevabını vermemiz gerekir. Programınızı çalıştırdığımızda eğer herhangi bir hata yoksa, direkt olarak konsol ekranı karşınıza çıkar (bkz. Şekil...). Bu ekranda programınız sizden bilgi istiyorsa onları girebilir ve sonuçları görebilirsiniz.



Figür 5: Programı çalıştırma

Hataların düzeltilmesi

Eğer programınızın derlenmesi sonucunda bir hatayla karşılaşırsanız, Visual Studio size programınızın önceki haliyle devam etmek isteyip istemediğinizi soracaktır. Bu soruya “Hayır” (No) yanıtını vererek programdaki hataları çözmemiz gerekmektedir. Bu gelen soruya hayır cevabını verdiğimizde programımızdaki hataların listesini görebiliriz. Bu liste şekil...’da gösterilmiştir. Eğer bu listeyi ekranınızda göremiyorsanız, şekil...’da gösterildiği gibi açabilirsiniz.

Bu listenin üst kısmında bulunan “Hatalar” (Errors), “Uyarılar” (Warnings) ve “Mesajlar” (Messages) düğmelerini kullanarak hata, uyarı ve mesajların gösterilmesini sağlayınız. Bu listede görünen hata, uyarı ve mesajlara çift tıklayarak hatanın bulunduğu satıra gidebilirsiniz. Bazen hatalar programın ayarlarından ya da genelinden kaynaklanmaktadır. Bu tip durumlarda hatanın hangi satırdan kaynaklandığı listede gösterilmeyebilir.

Bilmeniz gereken en önemli nokta hatanın önceki satırlardan kaynaklanabileceğini her zaman göz önünde bulundurmak gerektiğidir. Aşağıda belirtilen hataların dışında, çok sayıda hata alıyorsanız hataları almaya başladığınız satırdan önceki satırları kontrol edin. Bu satırlarda tırnak işaretini unutmuş, parantez kapatmayı unutmuş veya satırı “;” ile bitirmemiş olabilirsiniz.

Programınızı derlerken veya çalıştırırken hata vermemesine rağmen sonucun yanlış çıkmasına sebep olan hatalara mantık hataları denir. Böyle bir hatanız varsa programınızın yapısını dikkatle gözden geçirin. Bu tip hatalara daha sonra tekrar değinilecektir.

Aşağıda sıkça yapılan hatalar listelenmiştir. Programınız derlenirken verilen hataların dışında, programınız çalışırken ortaya çıkabilecek hatalar da mevcuttur. Bu hatalardan sıkça karşılaşılabilecekleriniz de bu listede yer almaktadır.

1. include teriminde hatalı sembol

Verdiği hata: Expected Filename...

“include” derleyici öncesi komutundan sonra dosya adı büyüktür, küçüktür sembolleri arasına yazılmalıdır.

2. hatalı tırnak işaretleri

Verdiği hata: unterminated string literal

C programlama dilinde metin dizeleri düz çift tırnak içerisinde bulunmalı ve her dize bir satırda tamamlanmalıdır. Bu hatayı yukarıdaki kurallar ihlal edildiğinde, yanlış tırnak işareti kullanıldığında veya tırnak işareti kapatılmadığında alırsınız. Kelime işlemcilerin tamamı yazım esnasında düz çift tırnakları şekilli tırnak açma ve kapatma işaretleriyle değiştirirler. Kopyala yapıştır işlemi yapıyorsanız bu hususa dikkat ediniz.

3. main fonksiyonu yok ya da birden fazla tanımlanmış

Verdiği hata: undefined symbol __main veya symbol __main is already defined

Bir C programında mutlaka “main” adından bir fonksiyon bulunmalıdır. main fonksiyonunuzun adını kontrol edin. C programlama dilinde bir fonksiyon ismi bir projede yalnızca bir kere

tanımlanabilir (özel durumlar hariç). Bu yüzden projenize ikinci bir dosya ekleyip yeni bir main fonksiyonu yazamazsınız.

4. WinMain fonksiyonu istenmekte

Verdiği hata: undefined symbol __WinMain

Visual Studio 2008'de pencere kullanan uygulamalarda yazabilirsiniz. Bu tarz bir uygulamanın projesi açtığımızda yapmanız gerekenler bu derste gösterilenlerden farklıdır. Böyle bir hatayı alıyorsanız proje tipinizi yanlış seçtiğiniz demektir. Yeni bir boş proje açarak yazdığımız kodları yeni projeye taşıyınız.

5. Noktalı virgül, virgül veya parantez kapatma eksik

Verdiği hata: Unexpected ... expecting ...

Belirtilen işareti koymayı unuttuğunuz anlamına gelir. Unutmayın bu hata her zaman hatanın size gösterdiği yerden önce gerçekleşir, bu yüzden hatadan önceki kısımları, gerekirse üstteki satırları da kontrol ediniz.

6. Değer atanmamış değişken

Verdiği uyarı: Variable ... is used without initializing

Belirtilen değişkene değer atamadan kullanmaya kalkıştığınızda ortaya çıkar. C programlama dilinde değişkenlerin ilk değerleri belirsizdir. Bu yüzden bir değişkeni kullanmadan önce ona değer atamanız gerekmektedir. Verilen mesaj uyarı mesajı olduğu için programınızın çalışmasını devam ettirebilirsiniz. Ancak ilk değer atanmamış bir değişken programınızın yanlış çalışmasına sebep olabilir.

7. Adres hatası

Verdiği hata: Sistem hatası veya access denied

Bu hatalar genelde "scanf" fonksiyonuyla ilgilidir. Bu fonksiyonda okutmak istediğiniz her değer için ek bir parametre ve gerekiyorsa adres operatörü (&) kullanılmalıdır.

Deney

1. Aşağıda verilen kodu derleyiniz. İstenilen değişiklikleri yapınız.

```
#include<stdio.h>
#include<stdlib.h>

int main() {

printf("Merhaba\n");
printf("Benim adim Ali\n");

system("pause");
return 0;}
```

Aşağıdaki çıktıyı verebilmesi için kodu nasıl değiştirmeliyiz?

Merhaba "Benim adim Ali"

2. Aşağıdaki kod klavyeden girilen iki tam sayıyı okuyor (x,y) ve toplamlarını buluyor (x+y).

```
#include<stdio.h>
#include<stdlib.h>

int main(){

int x,y;
printf("iki tam sayi giriniz:");
scanf("%d%d", &x,&y);
printf("%d + %d =%d \n",x + y);

system("pause");
return 0;}
```

- a) Girdi olarak 5 12 sayılarını kullanarak kodu çalıştırınız.
- b) Üç tam sayı (x,y,z) okuması için ve bu sayıların çarpımını bulması için (x*y*z) kodu nasıl değiştirmeliyiz? Girdi olarak 5 12 2 sayılarını kullanınız.