



## C++ - Modül 00

Namespaceler, sınıflar, member fonksiyonlar, stdio streamleri, initialization listeleri, static, const, ve bazı diğer basit şeyler

*Özet:*

*Bu doküman C++ Modül 00'in egzersizlerini içermektedir.*

*Versiyon: 8*

# İçindekiler

I	Giriş	2
II	Genel Kurallar	3
III	Egzersiz 00: Megafon	5
IV	Egzersiz 01: Benim Harika Telefon Rehberim	6
V	Egzersiz 02: Hayalinizdeki İş	8

# Bölüm I

## Giriş

*C++, Bjarne Stroustrup tarafından C programlama dilinin bir uzantısı olarak oluşturulan ve "Sınıf içeren C" olarak bahsedilen genel amaçlı bir programlama dilidir. (kaynak: [Wikipedia](#)).*

Bu modüllerin amacı size **Nesne Tabanlı Programlamayı** tanıtmaktır. Ve bu, C++ yolculuğunuzun başlangıç noktası olacak. Birçok dil OOP öğrenmeniz için önerilmekte. Eski dostunuz C'nin bir türevidiği için biz C++'ı seçtik. Bu karmaşık bir dil, işleri basit tutmak için kodunuz C++98 standardına uygun olacaktır. Modern C++'ın birçok açıdan çok farklı olduğunun farkındayız. Dolayısıyla, yetkin bir C++ geliştiricisi olmak istiyorsanız, 42 Common Core'dan sonra daha ileri gitmek size kalmış!

Adım adım yeni konseptler keşfedeceksiniz. Alıştırmaların karmaşıklığı giderek artacaktır.

# Bölüm II

## Genel Kurallar


C++ modülleri için sadece C++98 kullanabilirsiniz. Amacınız nesne tabanlı programlamanın temellerini öğrenmek. Modern C++'ın bir çok yönden daha farklı olduğunu biliyoruz fakat C++'ta uzmanlaşmak için modern C++'a daha sonra ihtiyacınız olacak. Bu sizin C++ yolculuğunuzun başlangıç noktası olacak, Common Core'dan sonra ne kadar ilerleyeceğiniz size kalmış!

- Header'da implement edilen herhangi bir fonksiyon (templateler hariç), ve korunmayan headerlar 0 puan anlamına gelir.
- Tüm çıktılar standart çıktıya olacak (stdout) ve yeni satırla bitecek (aksi belirtilmediği takdirde).
- Belirtilen dosya adları harfi harfine takip edilmelidir. Class, fonksiyon ve metod isimleri dahil.
- Hatırlatma: Artık C++'da kodluyorsunuz, C'de değil Bu yüzden:
  - Belirtilen fonksiyonların kullanımı **YASAKTIR**, ve kullanımları 0 ile cezalandırılacaktır. : **\*malloc, \*printf ve free.**
  - Standart kütüphanedeki tüm fonksiyonlar serbesttir. ANCAK, sadece bildiklerinizi korumak yerine C'deki fonksiyonların C++'a uyarlanmış hallerini kullanmayı denemek akıllıca olur. Ne de olsa yeni bir dil öğreniyorsunuz. Ve HAYIR, STL kullanacağınız yer belirtilene kadar kullanamazsınız. (modül 08'e kadar) Yani vector/list/map vb. include <algorithm>'i gerektirecek hiçbir şeyi oraya gelene kadar kullanamazsınız.
- Aslına bakarsak, yasak olan herhangi bir fonksiyon ya da mekanizma direkt olarak 0 ile cezalandırılacaktır.
- Aksi belirtilmediği takdirde, C++ keywordleri "using namespace" ve "friend" kullanımı yasaktır. Kullanımları direkt olarak -42 ile cezalandırılacaktır.
- Classla alakalı dosyalar ClassName.hpp ve ClassName.cpp olacaktır (aksi belirtilmediği takdirde)
- Yükleme klasörleri: ex00/, ex01/, ..., exn/.

- Örnekleri iyice okumanız gerekmektedir. Egzersiz açıklamasında açıkça belli olmayan gereklilikleri içerebilirler.
- Başlangıçtan beri öğrendiğiniz C++ araçlarını kullanmanız serbest olduğundan herhangi harici bir kütüphane kullanamazsınız. Siz sormadan, aynı zamanda C++11 ve türevleri de bu yasağa dahil.
- Önemli sayıda class'ı teslim etmeniz gerekebilir. Kendi favori text editörünüzü scriptleyene kadar bu sıkıcı görünebilir.
- Başlamadan önce tüm egzersizleri TAMAMEN okuyun! Yapın bunu.
- Kullanılacak derleyici: c++.
- Kodlarınız şu bayraklarla derlenecektir. `-Wall -Wextra -Werror`.
- Tüm include'lar diğerleri tarafından bağımsızca include edilebilmelidir. Include'lar bağlı oldukları diğer include'ları içermelidir.
- Merak ediyorsanız, belirli herhangi bir kodlama stili istemiyoruz. İstedığınız stilde yazabilirsiniz, kısıtlama yok. Şunu unutmayın ki sizi değerlendirecek arkadaşlarımız kodu okuyamazsa, notlandıramazlar da.
- Önemli bir şey: Özellikle belirtilmediği sürece, sizi bir program kontrol etmeyecek. Bu yüzden, egzersizleri istediğiniz şekilde yapmakta özgürsünüz. Ancak, her egzersizin kısıtlamasına dikkat edin, ve tembel OLMAYIN, sundukları BİRÇOK şeyi kaçırabilirsin.
- Yüklediğiniz dosyalarda bazı ekstra dosyalar olması sorun değil, , kodunuzu sizden istenenden daha fazla parçaya ayırabilirsiniz. Sonucunuz program tarafından belirlenmeyeceği sürece bu konuda özgür hissedebilirsiniz.
- Egzersiz açıklama dosyası kısa bile olsa sizden tam olarak ne beklendiğini anlamak için zaman harcayıp egzersizi mümkün olan en iyi şekilde yapmanıza değer.
- Odin'den, Thor'dan! Beynini kullan!!!

## Bölüm III

### Egzersiz 00: Megafon

	Exercise : 00
Megafon	
Turn-in directory : <i>ex00/</i>	
Files to turn in : Makefile, megaphone.cpp	
Forbidden functions : None	

Sadece herkesin uyanık olduğuna emin olmak için, aşağıdaki gibi davranan bir program yazınız.


```
$>./megaphone "shhhhh... I think the students are asleep..."
SHHHHH... I THINK THE STUDENTS ARE ASLEEP...
$>./megaphone Damnit " ! " "Sorry students, I thought this thing was off."
DAMNIT ! SORRY STUDENTS, I THOUGHT THIS THING WAS OFF.
$>./megaphone
* LOUD AND UNBEARABLE FEEDBACK NOISE *
$>
```



Alıştırmaları C++ tarzında çözün.

## Bölüm IV

# Egzersiz 01: Benim Harika Telefon Rehberim

	Exercise : 01
	My Awesome PhoneBook
	Turn-in directory : <i>ex01/</i>
	Files to turn in : <b>Makefile</b> , *.cpp, *.{h, hpp}
	Forbidden functions : <b>None</b>

80'lere ve onun inanılmaz teknolojilerine hoşgeldiniz! Rezil muhteşem bir telefon rehberi gibi çalışan bir program yazın.

İki sınıf geliştirmeniz gerekiyor:

- **PhoneBook**

- Bir kişiler dizisi içerecek.
- Aynı anda maksimum **8 kişi**yi depolayabilir. Eğer kullanıcı 9.'yu eklemeye çalışırsa en eskisinin yerine onu yerleştirin.
- Dinamik bellek kullanımının yasak olduğunu unutmayın.

- **Contact**

- Telefon rehberindeki kişi için kullanılacak.

Kodunuzda, telefon rehberi **PhoneBook** sınıfının bir örneği olarak başlatılmalıdır. Kişiler için de aynı şey. Bunların her biri, **Contact** sınıfının bir örneği olarak somutlaştırılmalıdır. Sınıfları istediğiniz gibi tasarlamakta özgürsünüz ancak bir sınıf içinde her zaman kullanılacak her şeyin özel olduğunu ve bir sınıf dışında kullanılabilecek her şeyin genel olduğunu unutmayın.



Intranet videolarını izlemeyi unutmayın.

Program başlangıcında, telefon rehberi boş olacak ve kullanıcı sadece 3 komut kullanabilecek. Program sadece ADD, SEARCH ve EXIT'i kabul etmelidir.

- **ADD:** Yeni bir kişi kaydet

- Kullanıcı bu komutu girerse, her seferinde bir alan olmak üzere yeni kişi bilgilerini girmesi istenir. Tüm alanlar tamamlandıktan sonra, kişiyi telefon defterine ekleyin.
- Kişi alanları şunlardır: ad, soyadı, takma ad, telefon numarası ve en karanlık sır. Kaydedilmiş bir kişinin boş alanları olamaz.

- **SEARCH:** Belirli bir kişiyi görüntüle

- Kaydedilen kişileri bir **4 sütun** listesi olarak görüntüleyin: dizin, ad, soyadı ve takma ad.
- Her sütun **10 karakter** genişliğinde olmalıdır. Bir çizgi karakteri ('|') onları ayırır. Metin sağa hizalı olmalıdır. Metin sütundan daha uzunsa, kısaltılmalı ve görüntülenebilir son karakter bir nokta ('.') ile değiştirilmelidir.
- Ardından, görüntülenecek girdinin dizinini kullanıcıdan tekrar isteyin. Dizin aralık dışında veya yanlışsa, ilgili bir tepki tanımlayın. Aksi takdirde, her satırda bir alan olacak şekilde kişi bilgilerini görüntüleyin.

- **EXIT**

- Program kapanır ve kişiler sonsuza dek kaybolur!

- **Başka girdiler yoksayılacaktır.**

Bir komut doğru bir şekilde yürütüldüğünde, program bir başkasını bekler. Kullanıcı EXIT girdiğinde sonlanır.

Yürütülebilir dosyanıza ilgili bir ad verin.




<http://www.cplusplus.com/reference/string/string/> ve tabii ki  
<http://www.cplusplus.com/reference/iomanip/>



## Bölüm V

### Egzersiz 02: Hayalinizdeki İş

	Exercise : 02
Hayalinizdeki İş	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <i>Makefile</i> , <i>Account.cpp</i> , <i>Account.hpp</i> , <i>tests.cpp</i>	
Forbidden functions : None	



*Account.hpp*, *tests.cpp* ve log dosyası intranetteki proje sayfasından indirilebilir.

Bugün *GlobalBanksters United*'daki ilk gününüz. İşe alım testlerini başarıyla geçtikten sonra (bir arkadaşınızın size gösterdiği birkaç *Microsoft Office* numarası sayesinde), geliştirme ekibine katıldınız. Ayrıca işe alım görevlisinin *Adobe Reader*'ı ne kadar hızlı yüklediğine şaşırdığınızı da biliyorsunuz. Bu küçük ekstra, fark yarattı ve tüm rakiplerinizi (diğer bir deyişle diğer adayları) yenmenize yardımcı oldu: başardınız!

Her neyse, müdürünüz size biraz iş verdi. İlk göreviniz kayıp bir dosyayı yeniden oluşturmaktır. Bir şeyler ters gitti ve bir kaynak dosyası yanlışlıkla silindi. Maalesef iş arkadaşlarınız *Git*'in ne olduğunu bilmiyor ve kodu paylaşmak için USB anahtarlarını kullanıyor. Bu noktada, burayı hemen terk etmek mantıklı olacaktır. Ancak kalmaya karar verdiniz. Meydan okuma kabul edildi!

Diğer geliştirici arkadaşlarınız size bir sürü dosya verdi. *tests.cpp* dosyasının derlenmesi, eksik dosyanın *Account.cpp* olduğunu ortaya çıkardı. Şanslısınız, *Account.hpp* header dosyası kaydedilmiş. Ayrıca bir günlük dosyası var. Belki **Account** sınıfının nasıl geliştirildiğini anlamak için kullanabilirsiniz.

*Account.cpp* dosyasını yeniden oluşturmaya başladınız. Sadece birkaç dakika içinde birkaç satırlık saf harika bir C++ kodu yazdınız. Birkaç başarısız derlemeden sonra

programınız testleri geçti. Çıktısı, günlük dosyasına kaydedilenle mükemmel bir şekilde eşleşti (**zaman damgaları hariç**, günlük dosyasına kaydedilen testler, işe alınmadan önce çalıştırıldığından mantıken farklılık gösterecektir).

Wow! Sen harikasın.



Destructorların çağrılma sırası, derleyicinize/işletim sisteminize bağlı olarak değişebilir. Yani destructorlarınız ters sırada çağrılabilir.



Bu modülü egzersiz 02'yi yapmadan da geçebilirsiniz.