



# OBJECT ORIENTED PROGRAMMING (OOP) WITH C++ EĞİTİMİ

2 GÜN



**Digital Vizyon**  
Akademi

[www.digitalvizyon.net](http://www.digitalvizyon.net)



## İçindekiler

|                                |   |
|--------------------------------|---|
| Eğitim Hakkında.....           | 3 |
| Neler Öğreneceksiniz? .....    | 3 |
| Ön Koşullar .....              | 4 |
| Kimler Katılmalı.....          | 4 |
| Outline .....                  | 5 |
| Introducing C++ .....          | 5 |
| Understanding C++ Memory ..... | 5 |
| Operator Overloading.....      | 5 |
| Template Classes .....         | 6 |

## Eđitim Hakkında

Object Oriented Programming (OOP) with C++ Eđitimi, katılımcılara C++ programlama dilini kullanarak nesne yönelimli programlama paradigmasını etkin bir şekilde uygulama yetenekleri kazandırır. Eđitim, karmaşıklığı yönetmek ve tekrarı azaltmak için güçlü bir araç olan nesne yönelimli programlama kavramlarının detaylı bir incelemesini içerir.

Eđitimde ilk olarak, nesne yönelimli programlama yaklaşımının temel taşları olan sınıflar ve nesnelere üzerinde durulur. Katılımcılar, C++'da sınıfların nasıl oluşturulacağını, nesnelere nasıl örnekleneceğini ve bu nesnelere nasıl manipüle edileceğini öğrenirler.

Kalıtım kavramı, eđitimin bir diđer önemli bileşenidir. Kalıtım, bir sınıfın başka bir sınıfın özelliklerini ve davranışlarını miras almasına olanak tanır. Katılımcılar, Object Oriented eđitiminde kalıtımın kod yeniden kullanımını nasıl kolaylaştırdığını ve bir sınıf hiyerarşisi oluşturmayı öğrenir. Daha sonra da uygulamaların daha modüler ve esnek hale gelmesini sağlarlar.

Polimorfizm, eđitim kapsamında incelenen bir başka kavramdır. Polimorfizm sayesinde, farklı sınıfların aynı arayüzü paylaşmasına olanak tanır. Bu, kodun daha genel ve soyut olmasına olanak tanır. Böylelikle daha az kod ile daha fazla işlevsellik sağlanır.

Encapsulation kavramı da bu eđitimin ayrılmaz bir parçasıdır. Encapsulation, veri ve işlevleri bir arada tutarak, dış dünyadan gizler. Bu, katılımcıların veri bütünlüğünü korumalarına ve kodun güvenliğini artırmalarını sağlar.

OOP'nin yanı sıra, katılımcılar nesne tabanlı tasarım prensiplerini de öğrenir. Sonrasında bu prensipler, katılımcılara OOP kavramlarını uygulamalarında nasıl entegre edeceklerini öğrenir. Bu, kodun daha sürdürülebilir, okunabilir ve bakımı kolay olmasını sağlar.

Eđitimin sonunda, katılımcılar OOP'nin avantajları ve dezavantajlarına dair bir anlayışa sahip olurlar. Bu, onların Object Oriented Programming yaklaşımını daha bilinçli ve etkili bir şekilde kullanmalarını sağlar.

Kısacası, Object Oriented Programming with C++ Eđitimi, nesne yönelimli programlama kavramlarını, prensiplerini ve C++ programlama dilinde uygulamalı örnekleri ile birleştirerek katılımcılara sağlam bir temel ve pratik beceriler kazandırır. Bu eđitim, katılımcıların yazılım geliştirme süreçlerini daha verimli, etkili ve sürdürülebilir hale getirmelerine yardımcı olur.

## Neler Öğreneceksiniz?

OOP C++ eđitiminde şunları öğrenebilirsiniz:

- Nesne tanımlama ve sınıf kavramları: Nesnelere ve sınıfları nasıl tanımlayacağınızı öğreneceksiniz.
- Kalıtım: Sınıflar arasında kalıtım ilişkisi kurma becerilerinizi geliştireceksiniz.



- Polymorphism: Farklı türde verilere uygulanacak farklı işlemleri tanımlama ve polymorphic fonksiyonları kullanma becerilerinizi geliştireceksiniz.
- Encapsulation: Verileri gizleme ve sınıf içindeki verilere erişimi kontrol etme becerilerinizi geliştireceksiniz.
- Nesne tabanlı tasarım prensipleri: Nesne yönelik programlama prensiplerini ve bu prensiplere göre yazılmış kodların yapısını ve işleyişini anlamaya çalışacaksınız.
- OOP tekniklerinin avantajları ve dezavantajları: Nesneye yönelik programlama tekniklerinin avantajlarını ve dezavantajlarını öğreneceksiniz.

## Ön Koşullar

OOP C++ eğitiminin ön koşulları şunlar olabilir:

- Temel programlama becerileri: Eğitim içeriğinin anlaşılması için programlama dillerinde temel kavramların bilinmesi gerekir.
- C++ diline hakimiyet: Eğitimin verildiği dil C++ olduğu için bu dille ilgili temel kavramların ve sintaksların bilinmesi faydalı olacaktır.
- Algoritma ve veri yapıları: Algoritma ve veri yapılarının temel kavramlarının bilinmesi, OOP kavramlarının anlaşılmasını kolaylaştırabilir.

Bu ön koşullar her eğitim verilen yerde aynı olmayabilir, ancak eğitim verilen yerin belirlediği ön koşulların yerine getirilmesi eğitim sürecinin verimli ve keyifli olmasını sağlar.

## Kimler Katılmalı

OOP C++ eğitimi şunlar için uygun olabilir:

- Programlama dillerinde ileri seviye becerilere sahip olanlar: Eğitimde verilen içerik daha önce öğrendiğiniz programlama dillerinin temel kavramlarının üzerine inşa edilecektir.
- C++ dili ile ilgilenenler: Eğitimin verildiği dil C++ olduğu için bu dille ilgilenenler faydalanacaktır.
- Nesneye yönelik programlamayı öğrenmek isteyenler: Eğitimin amacı nesneye yönelik programlama kavramlarını öğretmektir ve bu konuda ilgisi olanlar faydalanacaktır.
- Gelişmiş programlama becerilerini edinmek isteyenler: Eğitimde öğreneceğiniz OOP kavramları programlama becerilerinizi geliştirecektir.

Bu eğitim herkes için uygun olmayabilir, ancak programlama ile ilgilenen ve kendini geliştirmek isteyen herkes eğitimden yararlanabilir.



# Outline

## Introducing C++

- C++ Philosophies
- C++ Standards
- C++ Code Structure
- Using Visual Studio
- Creating Classes and Instantiating Objects
  - Declaring and Implementing classes
  - Understanding Encapsulation
  - Instantiating and Using Objects

## Understanding C++ Memory

- Heap vs Stack vs Global Memory
- Traditional Pointers
- References
- Modern Smart Pointers
- Constructors and Destructors
  - Implementing Simple Constructors
  - Do You Need a Copy Constructor?
  - Implementing Destructors
  - Using the Member Initialization List

## Operator Overloading

- Why Operator Overloading?
- Overloading Arithmetic Operators
- Overloading Comparison Operators
- Overloading the Assignment Operator
- Inheritance and Polymorphism
  - Creating a Derived Class
  - Polymorphism and Virtual Functions
  - Object Slicing
  - Heterogeneous Collections
  - Abstract Classes



## Template Classes

- Understanding Generic Programming
- Creating Templated Classes
- Using Templated Classes
- A Brief Glance at the C++ Standard Library
- Miscellaneous Tidbits
- Static Members and Static Constructors
- Preventing Copying
- Preventing Assignment
- C++ Evolution