



# **C++ PROGRAMLAMA EĞİTİMİ**

## **5 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 .....	4
Introduction and Overview.....	4
C++ Programming Building Blocks.....	4
The main function and standard I/O .....	5
Objects, constants, and references .....	5
Defining and calling functions .....	5
Decisions, loops, and logic.....	5
Arrays, pointers, and strings .....	5
Defining C++ Classes and Objects.....	5
Encapsulating higher-level data types.....	5
Declaring, accessing, and modifying objects.....	6
Overloading and templates .....	6
Separating interfaces and implementations.....	6
Extending Classes via Inheritance .....	6
Deriving new classes from existing classes.....	6
Utilizing polymorphic functions.....	6
Managing dynamic data .....	6
Standards and Portability .....	6

## Eđitim Hakkında

C++ programlama eđitimi, katılımcılara C++ diliyle uygulama geliştirme becerisi kazandırır. Veri yapıları, etkili veri yönetimini öğretir ve anlamak için kritiktir. Nesne yönelimli programlama, karmaşık yapıları yönetmeyi kolaylaştırır. Eđitim, ayrıca sistem programlama konularını da ele alır. Yüksek performanslı grafikler, görsel uygulamalar için önemlidir. Oyun geliştirme, eğlence endüstrisinde önemli bir rol oynar. C++ eđitimi, katılımcılara bu alanlarda uygulamalar geliştirmek için gereken bilgiyi verir.

Standart kütüphaneler, kod yazma sürecini hızlandırır ve kolaylaştırır. Fonksiyonlar, kodun yeniden kullanılabilir ve yönetilebilir olmasını sağlar. Diziler, birden çok veri ögesini düzenli bir şekilde saklar. Pointçiler, veriye doğrudan erişim sağlar ve performansı artırır. Sınıflar, nesne yönelimli programlamada temel yapı taşlarıdır. Nesnelere, gerçek dünya varlıklarının temsilleridir. Dosya işlemleri, veriyi kalıcı olarak saklama yeteneđini öğretir. Hata yönetimi, programların güvenilir ve stabil çalışmasını sağlar.

Sınıf ve nesne kalıtımı, kodun yeniden kullanımını artırır. Polymorphism, nesnelerin farklı tiplerde olmasına olanak tanır. Eđitim, C++'ın avantajlarını ve gücünü gösterir. C++'ın geniş kullanım alanları, eđitimi alacakları çeşitli kariyer yollarına yönlendirir. Sistem programlama, işletim sistemleri ve düşük seviyeli uygulamalar için temeldir. Grafik ve multimedya, görsel içerik üretiminde kullanılır. Veri madenciliđi, büyük veri setlerinden bilgi çıkarmak için kullanılır.

Katılımcılar, bu eđitimle karmaşık uygulamalar geliştirir. Modern C++ tekniklerini öğrenmek, kod kalitesini artırır. Performans iyileştirmeleri, uygulamaların daha hızlı çalışmasını sağlar. Eđitim, katılımcıları mevcut uygulamaları C++ ile geliştirmeye teşvik eder. Derinlemesine C++ bilgisi, katılımcılara güçlü bir temel sağlar. Eđitim, ayrıca problem çözme ve algoritma becerilerini geliştirir.

Kısacası, C++ programlama eđitimi, katılımcılara kapsamlı bir C++ bilgisi sağlar. Bu bilgi katılımcılara nasıl modern uygulama geliştireceklerini ve mevcutları uygulamaları nasıl iyileştireceklerini öğretir. Eđitim, programlama dünyasında başarılı bir kariyer için gereken becerileri kazandırır.

## Neler Öğreneceksiniz?

C++ programlama eđitimi süresince aşağıdaki konuları öğrenebilirsiniz:

- C++ programlama dilinin temel kavramlarını
- Veri tipleri, deđişkenler ve operatörleri
- Kontrol yapılarını (if, switch, for, while)
- Fonksiyonları
- Diziler, sıralı diziler ve sözlükleri
- Nesne yönelimli programlama kavramları (sınıflar, nesnelere, kalıtım, polymorphism)
- Standart kütüphanelerin kullanımı (input/output, string, vektör, liste, map vb.)
- Dosya işlemlerini



- Hata yönetimi ve exception handling
- Veri yapılarını (stacks, queues, linked lists, trees vb.) öğreneceksiniz.

Bu listede yer alan konular genellikle eğitim programına göre değişebilir ve diğer avans seviyedeki konular da eklenebilir.

## Ön Koşullar

C++ programlama eğitimine katılmak için genellikle aşağıdaki ön koşullar gerekmektedir:

- Temel programlama kavramlarının anlaşılması: Variables, loops, functions, conditional statements vb.
- Başka bir programlama dili (örneğin Python, Java, C) kullanmış olmak.
- Bilgisayarın kullanımı ve dosya sistemi hakkında temel bilgi sahibi olmak

Ancak bu ön koşullar eğitim verilen yerden yere değişebilir ve bazı eğitimler için daha az tecrübe gerekebilir. Eğitim veren kurumun websitesinden ya da eğitmenle direk olarak iletişime geçerek ön koşullar hakkında detaylı bilgi edinebilirsiniz.

## Kimler Katılmalı

C++ programlama eğitimine katılmak isteyen kişiler aşağıdaki profillere sahip olabilir:

- Temel programlama kavramlarını öğrenmek isteyen ve profesyonel olarak bu alanda kariyer yapmak isteyen kişiler.
- Başka bir programlama dili kullanmış olan ve C++ öğrenmek isteyen kişiler.
- Oyun, grafik ve multimedya uygulamaları, veri madenciliği, sistem programlama vb. gibi alanlarda uygulama geliştirmek isteyen kişiler.
- C++ ile yazılmış uygulamaları anlamak, geliştirmek veya optimize etmek isteyen kişiler.

Bu liste tamamen kişisel amaçlarına ve profesyonel hedeflerine göre genişletebilirsiniz. C++ programlama eğitimi size istediğiniz alanda beceriler kazandırarak kariyerinizi geliştirme fırsatı sunar.

## Outline

### Introduction and Overview

- Relating C, C++, Java, and C#
- The in-class development environment
- Other development environments

### C++ Programming Building Blocks



## The main function and standard I/O

- Displaying values and strings to cout
- Reading values from cin
- Formatting with stream manipulators

## Objects, constants, and references

- Declaring and initializing variables
- Utilizing C++ 11/14 new keywords
- Integer and floating-point data types
- Performing calculations and displaying results
- Utilizing references for efficiency and constants for safety

## Defining and calling functions

- Passing arguments to functions and returning values from functions
- Call-by-value vs. call-by-reference vs. call-by-address
- Scope, duration, and initial values variables

## Decisions, loops, and logic

- Making decisions with if/else
- bool vs. int logical values
- if/else statement “chains”
- Performing loops with while and for
- Implementing C++ range-based for loops
- Equality, relational, and logical operators

## Arrays, pointers, and strings

- Declaring and using arrays and pointers
- Storing strings in character arrays
- Accessing array elements via pointers
- Pointers vs. references
- Standard string class and functions

## Defining C++ Classes and Objects

### Encapsulating higher-level data types

- Public member functions and private data members
- Protected class members
- Constructors and destructors
- Self-reference: the pointer
- The class member operator (::)



## Declaring, accessing, and modifying objects

- Manipulating arrays of objects, pointers to objects and references to objects
- Invoking member functions
- const member functions
- Passing objects by value and by reference

## Overloading and templates

- Simplifying class interfaces
- Function signatures
- Overloading assignment (=) and insertion (<<)
- friend functions and classes
- Explicit copy/move construction
- Avoiding default assignment and copy construction
- Utilizing STL templates to define families of classes

## Separating interfaces and implementations

- How separation supports code reuse
- Building header files and code files

## Extending Classes via Inheritance

### Deriving new classes from existing classes

- Construction and destruction of derived objects
- Reusability via incremental extensions
- Base classes and derived classes

### Utilizing polymorphic functions

- Overriding virtual base-class member functions in derived classes
- Runtime lookup of functions through base-class pointers and references

### Managing dynamic data

- Allocating and deallocating memory with new and delete
- Handling errors with try and catch
- Avoiding memory leaks
- Utilizing lambdas and smart pointers

### Standards and Portability

- Applicability to Windows and UNIX/Linux
- ANSI/ISO C++ 11/14 and evolving standards