C++ Programlama Eğitimi

Eğitim Hakkında

C++ is a general purpose programming language that provides high-level abstraction without sacrificing run-time performance.

It is used for system and application development and according to many index, is one of the world's top 5 most popular programming language.

This course provides students with a comprehensive study of the C++ Programming Language. The course stresses the object paradigm including classes, inheritance, virtual functions...

Neler Öğreneceksiniz

- Compare the object vs the procedural approach to writing software
- Use correct object oriented terminology
- Define and use classes in a C++ Program
- Select the proper class protection mechanism
- Create and use abstract data types
- Derive classes using inheritance in C++
- Implement operator overloading
- Implement polymorphism by using virtual functions in a program
- Write programs utilizing the C++ i/o classes

Eğitim İçeriği

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 this 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