

## C++ Programming

#### **Audience C++ Programming Course**

The course <u>C++</u> Programming is intended for developers who want to learn programming in C++ and others who want to understand C++ code.

#### **Prerequisites Course C++ Programming**

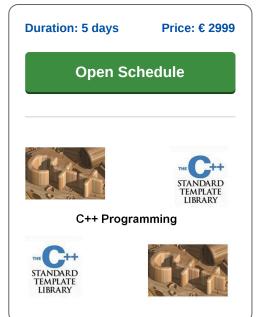
Knowledge of and experience with **C** programming is required to attend this course.

#### **Realization Training C++ Programming**

The theory is treated on the basis of presentation slides and is interspersed with exercises. Illustrative demos are used to clarify the discussed concepts. The course material is in English.

#### **Certification C++ Programming**

Participants receive an official certificate C++ Programming after successful completion of the course.



### **Content C++ Programming**

In the course C++ Programming participants learn to program in the C++ language. In the course the latest version of the C++ standard is used.

#### **Differences C and C++**

First the differences between C and C++ are discussed concerning variable declarations, formatted output with the stream IO library, namespaces, function overloading and default function parameters.

#### References

Subsequently the new C++ reference variables are discussed. Attention is paid to both Lvalue and Rvalue references.

#### C++ Classes

An important element of the course is the C++ class concept and C++ implementation of object-oriented principles such as abstraction and encapsulation. Attention is paid to dynamic memory allocation with new and delete and the role of assignment operators and copy and move constructors. Also special features of classes such as statics, friends and iterators are discussed.

#### **Inheritance and Polymorfisme**

Next the object-oriented principles of inheritance and polymorphism are part of the subject matter. This includes the concepts of virtual functions, v-tables, dynamic binding and abstract classes.

#### **Operator Overloading**

C++ has the option to give existing operators a different meaning and this phenomenon is discussed in the module operator overloading.

#### **Templates en Standard Template Library**

Attention is paid to important features of the standard C++ library like the String class and the base concepts of C++ templates and the Standard Template Library (STL).

#### **Exception Handling**

Finally exception handling and how this is implemented in C++ is addressed. A follow up course for the course C++ Programming is **Advanced C++ Programming**.

Tel.: +31 (0) 30 - 737 0661



# **Modules C++ Programming**

Intro C++ C++ Time Line C++ Time Line C+- Time Line C Comments in C++ Auto Keyword C References and Pointers Namespace std Deduction with decitype Intitialization References and Pointers References and Pointers References and Pointers References and Pointers Namespace std Null Pointer Constant Reference to Constant Reference to Constant Reference to Constant Reference to Constant References as Return Values References as Return Values References as Return Values Returning Native Returning Native Returning Native Returning Native Returning Reference to Globs Returning Reference to Globs Returning Reference to Globs Revalue References Comparing Reference Types Range based for loop Rodule 4: Classes Const Qualifier Revalue Reference Const Qualifier Revalue Reference Const Qualifier Revalue Reference References Comparing Reference to Globs Revalue Reference Returning Reference to Globs Revalue Reference Sometime Value Returning Reference to Globs Revalue Reference References Comparing Reference Types Revalue Reference Sometime Value Reference Comparing Reference Types Revalue Reference References References References Comparing Reference References References References References to Globs Returning Native Returning Reference References Comparing Reference Returning Reference References Comparing Reference References Comparing Reference Returning Reference Returni	
Auto Keyword References and Pointers Rvalues and Rvalues in C Deduction with decitype Initialization Rvalues and Rvalues in C Deduction with decitype Initialization Rvalues and Rvalues in C Proceedings of the Reference to Constant Initialization Rvalues and Rvalues in C Proceedings of the Reference to Constant Reference to Constant Reference to Constant Reference to Comparison Parameter Passi Ariable Declaration Storope Resolution Operator Using Revenor and Directive References as Return Values Scope Resolution Operator Using Revenor and Directive Returning References as Return Values Scope Resolution Operator Using Revenor and Directive Returning Reference of Globic Returning References as Return Values Scope Resolution References as Return Values References to Globic Returning Reference Usage Returning Reference Usage Returning Reference Usage Revenues Types Range based for loop Comparison Parameter Passis References to Comparing Reference Usage References to Comparing Reference Usage References and Objects Classes in C++ Dynamic Memory Allocation Module 5: Inheritance Inheritance Inheritance Dynamic Arrays Derived Classes in C++ Classes Declaration Classes with Pointer Data Assignment Operator Redefining Member Functions Classes Inheritance Pointer Conversions Virtual Functions Passing Objects Pointer Conversions Virtual Functions Passing Objects Pointer Pointer Passing Objects Pointer Pointer Pointer Pointer Passing References to Objects Pointer	
Namespace std Deduction with decitype Initialization Null Pointer Constant Standard Input Stream Null Pointer Constant Strongly Types Enums Formatted Output Variable Scope Variable Declaration Namespaces Variable Scope Variable Scope Variable Scope Variable Declaration Namespaces Variable Declaration Namespaces Variable Declaration Namespaces Variable Scope Variable Declaration Variable Scope Variable Scope Variable Scope Variable Scope Variable Scope Variable V	
Namespace std Deduction with decitype Initialization Null Pointer Constant Standard Input Stream Null Pointer Constant Null Pointer Constant Strongly Types Enums Passing Reference to Constant Reference to Constant Reference to Constant Passing References Passing References Comparison Parameter Passi Namespaces Namespaces Namespaces Namespaces Namespaces Namespaces References as Return Values Returning Native Returning Reference to Globs Revalue References Revalue References Revalue Reference Socones Revalue Revalue Revalue Revalue Revalue Revalue Revalue Revalue Revalue R	
Standard Input Stream In and Strings Strongly Types Enums Strongly Types Enums Passing References Comparison Parameter Passi References as Return Values Returning Value Returning Reference to Globa Returning Value	
cin and Strings Formatted Output Variable Scope References as Return Values Returning Natue Returning Reference to Globz Returning Reference to Globz Returning Reference Spress Value References Scope Nature Returning Reference Types Variable Functions Variable Scope Variable Functions Variable Scope Variable Function Arguments Variable Vasage Variable Reference Spress Variable Reference Spress Variable Reference Usage Variable Scope Variable Functions Variable	
cin and Strings Formated Output Variable Scope References as Return Values Returning Nalue Returning Reference to Globz Returning Reference to Globz Returning Reference Subseture Value References as Return Values Returning Reference to Globz Returning Reference to Globz Returning Reference Types Comparing Reference Types Constitution Variable Scope Module 4: Classes  Module 5: Dynamic Memory Allocation Module 6: Inheritance Classes and Objects Classes and Objects Classes and Objects Classes on C++ Class Declaration Classes with Pointer Data Assignment Operator Constructor and Destructor Uniform Initialization Constructor and Destructor Value Reference Spee Variable Scope Variable Scope Variable Scope Variable Scope Variable Scope Variable Scope Reterence Variable Variable Scope Variable Scope Variable Variable Variable Scope Variable Variab	
Formatted Output Variable Declaration Namespaces Scope Resolution Operator Inline Functions Default Function Arguments Overloading Functions Scope Agesolution Operator Inline Functions Default Function Arguments Overloading Functions Storage Classes Comparing Reference to Globe Returning Reference to Grow and Returning Reference to Globe Returning Objects Polymorphism Dornatic Base - Derived Class Convertive Glass Convertive Gl	
Variable Declaration Scope Resolution Operator Using keyword and Directive Block Usage Default Function Arguments Default Function Arguments User Defined Literals Storage Classes Overloading Functions Storage Classes Companing Reference to Globa Rvalue References Companing Reference Usage Returning Reference Usage Rvalue References Companing Reference Usage Rvalue References Companing Reference Usage Rvalue References Ormpaning Reference Storage Rvalue References Ormpaning Reference Storage Rvalue References Ormpaning References Usage Rvalue References Usage Inheritance Derived Class Inheritance Derived Class Inheritance Derived Class Constructor Virtual Functions Rvalue References Usage Rvalue References Usage Rvalue References Rva	na
Using keyword and Directive Returning Ivalue Returning Reference to Globe Default Function Arguments Diverloading Function Storage Classes Overloading Functions Storage Classes Comparing Reference Types Range based for loop  Module 4: Classes Module 5: Dynamic Memory Allocation Module 6: Inheritance Classes and Objects Classes and Objects Classes in C++ Dynamic Arrays Classes Declaration Classes Sections Constructor and Destructor Uniform Initialization Chained Assignment Problem Classes Innetiation Copy Constructors Caps Inheritance Copy Constructors Constructor and Sources Files Classes Innetiance Copy Constructors Copy Constructors Copy Constructors Copy Constructors Copy Constructors Constant Objects Constant Objects Constant Objects Constant Objects Constant Objects Constant Objects Constructors Copy Constructors Copy Constructors Copy Constructors Copy Constructors Copy Constructor Copy C	
Inline Functions Default Function Arguments Default Function Arguments User Defined Literals Storage Classes Range based for loop  Module 4: Classes  Module 5: Dynamic Memory Allocation Module 6: Inheritance Classes and Objects Inheritance Classes in C++ Dynamic Arrays Classes in C++ Dynamic Arrays Classes with Pointer Data Class Sections Constructor and Destructor Uniform Initialization Class Inheritance Class Inheritance Class Sections Casses Intervention Chained Assignment Problem Class Information Class Independent of Chained Assignments Class Inheritance Derived Class Convertion Assignment Operator Self-Assignment Problem Derived Class Convertion Constructor and Destructor Self-Assignment Problem Derived Class Convertion Constructor and Destructor Chained Assignments Assignment and Initialization Copy Constructors Virtual Functions Advantages Access Functions References to private Data Returning Objects Passing References to Objects Move Constructor Constant Objects Move Assignment Operator Abstract Classes Members Constant Objects Move Assignment Operator Abstract Classes Multiple Inheritance Virtual Functions Abstract Classes Multiple Inheritance Virtual Derivation Module 7: Operator Overloading Delegating Constructors Virtual Derivation  Module 9: Templates  Deverloading for Numeric Types Complex Type Example Throwing Exceptions Throwing Exceptions Throwing Exceptions Template Parameter List Class Template Parameter List Class Template Parameter List Class Template Parameter Scope Template Scope S	
Default Function Arguments Overloading Functions Storage Classes Comparing Reference Types Rvalue Reference Types Inheritance Derived Classes in C++ Class Exemptole Derived Classes in C++ Class Hierarchy Redefining Member Functions Base - Derived Class Conver Redefining Member Functions Base - Derived Class Conver Redefining Member Functions Pointer Conversions Virtual Functions Pointer Conversions Virtual Functions Pointer Conversions Virtual Function Table Porention Table Porention Types References to Objects Perfect Forwarding Perfect	al
Storage Classes const Qualifier Reference Types Range based for loop  Module 4: Classes  Module 5: Dynamic Memory Allocation  Module 6: Inheritance  Classes and Objects  Classes in C++  Classes in C++  Classes in C++  Classes with Pointer Data  Classes with Pointer Data  Classes ections  Casses with Pointer Departor  Constructor and Destructor  Uniform Initialization  Chained Assignment Problem  Class Access Functions  Assignment Assignments  Base - Derived Class Constructors  Uniform Initialization  Chained Assignments  Base - Derived Class Constructors  Uniform Initialization  Chained Assignments  Base - Derived Class Constructors  Uniform Initialization  Chained Assignments  Base - Derived Class Constructors  Class Implementation  Copy Constructors  Advantages Access Functions  References to private Data  Returning Objects  Returning Objects  Passing References to Objects  Wove Assignment Operator  Move Assignment Operator  Pure Virtual Function Table  Pure Virtual Functions  Abstract Classes  Multiple Inheritance  Citasses with Pointer  Module 8: Exception Handling  Module 9: Templates  Departor Overloading  Module 9: Templates  Template Functions  Template Punctions  Template Punctions  Template Parameter List  Class Template Parameter Scope  Template Parameter Scope	-
Range based for loop  Const Qualifier  Module 4: Classes  Module 5: Dynamic Memory Allocation  Module 6: Inheritance  Inheritance  Classes and Objects  Classes and Objects  Classes in C++  Dynamic Arrays  Classes with Pointer Data  Assignment Operator  Constructor and Destructor  Constructor and Destructor  Constructor and Destructor  Constructor and Destructor  Constructor and Objects  Classes Sections  Constructor and Destructor  Constructor Assignment and Initialization  Constructor Assignment and Initialization  Copy Constructors  Class Implementation  Copy Constructors  Constructor  Copy Constructors  Passing Objects  Returning Objects  Polymorphism  Passing References to Objects  Move Constructor  Passing References to Objects  Move Constructor  Move Assignment Operator  Abstract Classes  Multiple Inheritance  Virtual Functions  Abstract Classes  Multiple Inheritance  Virtual Derivation  Module 7: Operator Overloading  Module 8: Exception Handling  Module 9: Templates  Departor Overloading  Complex Type Example  Overloading Rules  Overloading Rules  Overloading Rules  Overloading Rules  Overloading Restrictions  Not Overloadable Operators  Multiple catch Handlers  Template Parameter List  Class Template Parameter Scope  Template Class Overload  Numeric Class Overloading  Overloading Operator  Module 10: STL	
Module 4 : Classes  Module 5 : Dynamic Memory Allocation  Module 6 : Inheritance  Classes and Objects  new and delete Operators  Dynamic Arrays  Classes in C++  Class Declaration  Classes with Pointer Data  Assignment Operator  Constructor and Destructor  Uniform Initialization  Chained Assignments  Base - Derived Class Constructors  Uniform Initialization  Chained Assignments  Base - Derived Class Constructors  Uniform Initialization  Chained Assignments  Base - Derived Class Converses Class Implementation  Assignment and Initialization  Copy Constructors  Passing Objects  References to private Data  Returning Objects  References to private Data  Returning Objects  Passing References to Objects  Move Constructor  Constant Objects  Move Assignment Operator  Module 7 : Operator Overloading  Module 8 : Exception Handling  Delegating Constructors  Module 7 : Operator Overloading  Overloading Rules  Overloading Restrictions  Not Overloadable Operators  When not to Overloading  Operator Sa Friend  Module 10 : STL	
Classes and Objects Classes in C++ Class Declaration Classes with Pointer Data Class Sections Constructor and Destructor Uniform Initialization Class Assignment Problem Constructor Initialization Class Berearchy Class Serving Constructor Self-Assignments Constructor Self-Assignments Constructor Self-Assignments Constructor Class Initialization Chained Assignment Problem Chained Assignments Constructor Self-Assignments Constructor Self-Assignments Constructor Class Conversions Class Implementation Chained Assignment Problem Chained Assignments Conversions Conversions Core Core Core Conversions Core Core Core Conversions Core Core Core Core Conversions Core Core Core Core Core Core Core Core	
Classes in C++ Class Declaration Classes with Pointer Data Classes with Pointer Data Classes Sections Constructor and Destructor Constructor and Destructor Uniform Initialization Chained Assignment Problem Class Assignment and Initialization Copy Constructors Class Implementation Copy Constructors Copy Constructors Copy Constructors Class Implementation Copy Constructors Copy Copy Copy Copy Copy Copy Copy Copy	
Class Declaration Class Sections Constructor and Destructor Constructor and Destructor Self-Assignment Problem Class Sections Constructor and Destructor Self-Assignment Problem Constructor and Destructor Self-Assignment Problem Chained Assignment S Base - Derived Class Constructors Chained Assignment and Initialization Copy Constructors Class Implementation Copy Constructors Copy Copy Copy Copy Copy Copy Copy Copy	
Class Sections Constructor and Destructor Constructor and Destructor Constructor and Destructor Self-Assignment Problem Chained Assignments Chained Assignments Copy Constructors Copy Construct	
Constructor and Destructor Uniform Initialization Chained Assignments Assignments Assignment and Initialization Class Implementation Copy Constructors Advantages Access Functions References to private Data Returning Objects References to private Data Returning Objects References to private Data Returning Objects References to Objects Robjects Move Constructor Constant Objects Move Assignment Operator Move Assignment Operator Perfect Forwarding Delegating Constructors  Module 7: Operator Overloading  Overloading Rules Overloading Restrictions Not Overloadding Robjects Not Overloadding Not	
Uniform Initialization Header and Sources Files Class Implementation Advantages Access Functions Passing Objects Passing Objects Passing References to Objects Move Constructor Pure Virtual Function Table Pure Virtual Functions Abstract Classes Multiple Inheritance Virtual Derivation  Module 7: Operator Overloading Delegating Constructors  Module 8: Exception Handling Module 9: Templates  Overloading for Numeric Types Memory Exhaustion Handling Complex Type Example Overloading Rules Overloading Rules Overloading Restrictions Anot Overload Restrictions Anot Overload Anot	5
Assignment and Initialization Class Implementation Class Implementation Copy Constructors Passing Objects References to private Data Returning Objects Returning Objects Passing References to Objects Returning Objects Passing References to Objects Static Members Constant Objects Move Constructor Move Constructor Pure Virtual Function Table Pure Virtual Functions Abstract Classes Member Objects Perfect Forwarding Pelegating Constructors Module 7: Operator Overloading Module 8: Exception Handling Module 9: Templates  Departor Overloading Overloading Functions Complex Type Example Overloading Rules Overloading Restrictions Not Overload Departors Multiple catch Handlers Catch Handlers Catch Handlers Catch Order Catch Module 10: STL	
Class Implementation Advantages Access Functions References to private Data References to private Data References to private Data Returning Objects Returning Objects Passing References to Objects Passing References to Objects Wirtual Function Table Pure Virtual Function Table Pure Virtual Functions Abstract Classes Member Objects Move Assignment Operator Module 7: Operator Overloading Perfect Forwarding Delegating Constructors  Module 7: Operator Overloading  Deperator Overloading Pexception Handling in C++ What are Templates? Complex Type Example Deverloading Rules Deverloading Rules Deverloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Deperator Auditiple catch Handlers Deperator Service Returning Objects Perfect Forwarding Module 8: Exception Handling Module 9: Templates  What are Templates? When are Template Punctions Template Functions Template Punctions Template Parameter List Class Template Sepcialization Template Parameter List Class Templates  Not Overloadable Operators When not to Overload Template Array Class Template Parameter Scope Template Param	sion
Advantages Access Functions References to private Data Returning Objects References to private Data Returning Objects Returning Objects Returning Objects Returning Objects Passing References to Objects Rove Constructor Rove Operator Objects Move Assignment Operator Module 7: Operator Overloading Perfect Forwarding Delegating Constructors  Module 8: Exception Handling Perplates  Departor Overloading Depart	
References to private Data References to private Data References to private Data References to Objects Rossing References to Objects Rowe Constructor Rowe Constructor Rowe Assignment Operator References to Objects Rowe Assignment Operator References to Objects Rowe Assignment Operator Rowe Rowe Assignment Operator Rowel Assignment Operator Rowell Assignment Operator Rowell Assignment Operator Rowell Assignment Operator Rowe	
this Pointer static Members	
Move Constructor Constant Objects Move Assignment Operator Move Assignment Operator Module 7: Operator Overloading  Operator Overloading Overloading Friends  Module 8: Exception Handling  Module 9: Templates  Exception Handling  Module 9: Templates  What are Templates?  Template Functions  Template Functions  Template Parameter List  Overloading Rules Overloading Restrictions  Not Overload Perators  When not to Overload  Multiple Inheritance  Virtual Derivation  Module 9: Templates  What are Templates?  Template Functions  Template Functions  Template Parameter List  Class Template Parameter List  Class Template Parameter Scope  When not to Overload  Template Array Class  Numeric Class Overloading  Exceptions Array Class  Catch Order  Unary Overloading Operator  Module 10: STL	
More Assignment Operator Member Objects Perfect Forwarding Delegating Constructors  Module 7: Operator Overloading Operator Overloading Operator Overloading Overloading Friends  Memory Exhaustion Handling Overloading Rules Overloading Restrictions Overloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Overloading Operators When not to Overloading Operator Overloading Overloading Restrictions Overloading Operators Overloading Class Overloading Overloading Operators as Friend Overloading Operator Overloading Operator Overloading Operator Overloading Operator	
Member Objects Friends  Perfect Forwarding Delegating Constructors  Module 7: Operator Overloading  Module 8: Exception Handling  Module 9: Templates  Exception Handling in C++ What are Templates?  Overloading for Numeric Types  Complex Type Example Overloading Rules Overloading Restrictions  Overloading Restrictions Not Overloadable Operators  Multiple catch Handlers Template Parameter List Class Template Parameter Scope Template Array Class Template Parameter Scope Template Array Class Template Parameter Scope Template Array Class Numeric Class Overloading Operators as Friend Unary Overloading Operator  Module 10: STL  Multiple Inheritance Virtual Derivation  Multiple Inheritance Virtual Derivation  Module 9: Templates  What are Templates? Template Functions Template Functions Template Parameter List Class Template Parameter Scope Template Array Class Template Class Statics Inclusion Compilation Model Templates and Friends	
Delegating Constructors   Virtual Derivation	
Module 7 : Operator Overloading Operator Overloading Operator Overloading Overloading for Numeric Types Overloading for Numeric Types Overloading Faxeuption Overloading Rules Overloading Rules Overloading Restrictions Overloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Numeric Class Overloading Operators Operators as Friend Overloading Operator Overloading Restrictions Overloading Restrictions Overloadable Operators Operator O	
Operator Overloading Overloading for Numeric Types Overloading for Numeric Types Complex Type Example Overloading Rules Overloading Restrictions Overloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Operators as Friend Unary Overloading Operator  Module 10 : STL  What are Templates? What are Templates?  What are Templates?  What are Templates?  Template Functions  Template Functions  Template Parameter List Class Templates  Template Parameter Scope  Template Array Class  Template Function Statics  Template Class Statics  Template Class Statics  Templates and Friends	
Overloading for Numeric Types  Complex Type Example  Overloading Rules  Overloading Restrictions  Not Overloadable Operators  When not to Overloading  Numeric Class Overloading  Numeric Class Overloading  Operators as Friend  Unary Overloading Operator  Memory Exhaustion Handling  Template Functions  Template Specialization  Template Parameter List  Class Templates  Template Parameter Scope  Template Parameter Scope  Template Function Statics  Template Functions	
Complex Type Example Overloading Rules Overloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Operators as Friend Unary Overloading Operator  Module 10 : STL  Template Specialization Template Parameter List Class Templates Template Parameter Scope Template Array Class Template Function Statics Template Class Statics Template Class Statics Inclusion Compilation Model Templates and Friends	
Overloading Rules Overloading Restrictions Overloadable Operators Not Overloadable Operators When not to Overload Numeric Class Overloading Operators as Friend Unary Overloading Operator  try Block catch Handlers Multiple catch Handlers Template Parameter Scope Template Parameter List Class Template Parameter Scope Template	
Overloading Restrictions  Not Overloadable Operators  When not to Overload  When not to Overload  When not to Overload  Summeric Class Overloading  Operators as Friend  Unary Overloading Operator  Module 10 : STL  Class Templates  Template Parameter Scope  Template Function Statics  Template Class Statics  Template Class Statics  Inclusion Compilation Model  Templates and Friends	
Overloading Restrictions  Not Overloadable Operators  When not to Overload  When not to Overload  When not to Overload  Summeric Class Overloading  Operators as Friend  Unary Overloading Operator  Module 10 : STL  Class Templates  Template Parameter Scope  Template Function Statics  Template Class Statics  Template Class Statics  Inclusion Compilation Model  Templates and Friends	
When not to Overload  Numeric Class Overloading  Deperators as Friend  Unary Overloading Operator  Module 10 : STL  Template Function Statics  Template Class Statics  Template Class Statics  Inclusion Compilation Model  Templates and Friends	
Numeric Class Overloading  Departors as Friend  Unary Overloading Operator  Module 10 : STL  Exceptions Array Class catch Order throw List  Template Class Statics Inclusion Compilation Model Templates and Friends	
Numeric Class Overloading Exceptions Array Class Operators as Friend catch Order Inclusion Compilation Model Unary Overloading Operator Templates and Friends  Module 10 : STL	
Unary Overloading Operator throw List Templates and Friends  Module 10 : STL	
Unary Overloading Operator throw List Templates and Friends  Module 10 : STL	
Standard Template Library	
STL Core Components	
STL Library Components	
STL Containers	
Vector Container	
Deque Container	
List Container	
STL Iterators	
STL Algorithms STL Allocators	

Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online