SpriralTrain BV
Standerdmolen 10, 2e verdieping.
3995 AA Houten

spiraltrain@gmail.com
www.spiraltrain.nl
Tel.: (0)30 737 0661
Fax: (0)30 737 1050

KvK nr. : 61618039
BTW nr. : NL854415798B01
IBAN : NL58RABO0156271540
BIC : RABONL2U

PRG300: Cursus C++ Programmeren

Code: PRG300 **Duur:** 5 dagen **Prijs:** € 2250

Doelgroep C++ Opleiding

De cursus C++ Programmeren is bedoeld voor developers die in C++ willen leren programmeren en anderen die C++ code willen begrijpen.

Voorkennis voor de Cursus C++

Om aan deze cursus te kunnen deelnemen is kennis van en ervaring met programmeren in C vereist.

Uitvoering Training Cursus C++

De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met oefeningen. Illustratieve demo's worden gebruikt om de behandelde begrippen te verduidelijken. De cursustijden zijn van 9.30 tot 16.30.

C++ Programming

Certificering C++

De deelnemers krijgen na het goed doorlopen van de training een officieel certificaat C++ Programmeren.

Inhoud van de Training C++

In de cursus C++ Programmeren leren de deelnemers programmeren in de C++ programmeertaal. In de cursus wordt de nieuwste C++ 17 standaard gebruikt. Eerst worden de verschillen tussen C en C++ besproken voor wat betreft declaraties van variabelen, formatted output met de stream IO library, namespaces, function overloading en default function parameters. Vervolgens wordt aandacht besteed aan de nieuwe C++ reference variabelen. Zowel Lvalue als Rvalue references worden besproken. Een belangrijk onderdeel van de cursus is het C++ class concept en de C++ implementatie van object georiënteerde principes als abstraction en encapsulation. Ook wordt aandacht besteed aan dynamische geheugen allocatie met new en delete en de rol van assignment operators en copy en move constructors. Ook speciale kenmerken van classes, zoals statics, friends en iterators worden besproken. Vervolgens staan ook de object georiënteerde principes van inheritance en polymorfisme op het programma. Daarbij komen de concepten van virtual functions, v-tables, dynamic binding en abstract classes ter sprake. In C++ is het mogelijk om standaard operatoren een andere betekenis te geven en dit fenomeen wordt besproken in de module operator overloading. Vervolgens komen belangrijke kenmerken van de C++ standard library aan bod zoals de String class en de principes van C++ templates en de Standard Template Library (STL). Tot slot wordt aandacht besteed aan exception handling en hoe dit is geïmplementeerd in C++.

Module 1 : Intro C++	**Module 2 : Variables and Types**	**Module 3 : References**
Intro C++	Standard Types	References
C++ TimeLine	Type Inference	Reference Initialization
Comments in C++	Auto Keyword	References and Pointers
Namespace std	Deduction with decltype	Rvalues and Rvalues in C
Output and Error Stream	Initialization	Rvalues and Rvalues in C++
Standard Input Stream	Null Pointer Constant	Reference to Constant
cin and Strings	Strongly Types Enums	Passing References
Formatted Output	Variable Scope	Comparison Parameter Passing
Variable Declaration	Namespaces	References as Return Values
Scope Resolution Operator	Using keyword and Directive	Returning lvalue
Inline Functions	Block Usage	Returning Reference to Global
Default Function Arguments	User Defined Literals	Rvalue References
Overloading Functions	Storage Classes	Comparing Reference Types
Range based for loop	const Qualifier	Rvalue Reference Usage

Module 4 : Classes	**Module 5 : Dynamic Memory Allocation**	**Module 6 : Inheritance**
Classes and Objects	new and delete Operators	Inheritance
Classes in C++	Dynamic Arrays	Derived Classes in C++
Class Declaration	Classes with Pointer Data	Class Hierarchy
Class Sections	Assignment Operator	Redefining Member Functions
Constructor and Destructor	Self-Assignment Problem	Derived Class Constructors
Uniform Initialization	Chained Assignments	Base – Derived Class Conversion
Header and Sources Files	Assignment and Initialization	Pointer Conversions
Class Implementation	Copy Constructors	Virtual Functions
Advantages Access Functions	Passing Objects	Polymorphism
References to private Data	Returning Objects	Dynamic Binding
this Pointer	Passing References to Objects	Virtual Function Table
static Members	Move Constructor	Pure Virtual Functions
Constant Objects	Move Assignment Operator	Abstract Classes
Member Objects	Perfect Forwarding	Multiple Inheritance
Friends	Delegating Constructors	Virtual Derivation

Module 7 : Operator Overloading	Module 8 : Exception Handling	Module 9 : Templates
Operator Overloading Overloading for Numeric Types Complex Type Example Overloading Rules Overloading Restrictions Not Overloadable Operators When not to Overload Numeric Class Overloading Operators as Friend Unary Overloading Operator	Exception Handling in C++ Memory Exhaustion Handling Throwing Exceptions try Block catch Handlers Multiple catch Handlers Template Array Class Exceptions Array Class catch Order throw List	What are Templates? Template Functions Template Specialization Template Parameter List Class Templates Template Parameter Scope Template Function Statics Template Class Statics Inclusion Compilation Model 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