

PRG300 : C++ Programmeren

Code : PRG300

Duur : 5 dagen

Categorie : Programming

Doelgroep :

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

Voorkennis :

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

Uitvoering :

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.



C++ Programming



Inhoud :

In de cursus C++ Programmeren leren de deelnemers te programmeren in de C++ programmeertaal. 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. Een belangrijk onderdeel van de cursus is het C++ class concept en de C++ implementatie van object georienteerde 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 constructors. Ook speciale kenmerken van classes, zoals statics, friends en iterators worden besproken. Vervolgens staan ook de object georienteerde 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++

- C++ Features
- Comments in C++
- I/O Stream Library
- Output and Error Stream
- Input Stream
- Multiple Input Values
- IO Stream Function get
- Formatted Output
- Function Prototypes
- Default Function Arguments
- Inline Functions
- Macro Versus Inline Function
- Overloading Functions
- Variable Declaration
- Namespaces
- Scope Resolution Operator

Module 2 : References

- References
- Reference Initialization
- References and Pointers
- Reference to Constant
- Passing References
- Comparison Parameter Passing
- References as Return Values
- Potential Error
- Potential Problem
- Returning Reference to Global

Module 3 : Classes

- Classes in C++
- New Data Type in C
- Class Declaration
- Formal Class Declaration Syntax
- Class Sections
- Constructor
- Destructor
- Class Implementation
- Member Functions
- Class Usage
- Calling Member Functions
- Reading Data Members
- Setting Data Members
- Header and Sources Files
- Advantages Access Functions
- References to private Data
- Constant Objects
- Member Objects
- Composition Example

Module 4 : Dynamic Memory Allocation

- Dynamic Memory Allocation
- new Operator
- delete Operator
- Dynamic Arrays
- Classes with Pointer Data
- Assignment Operator
- Self-Assignment Problem
- this Pointer
- Using this Pointer
- Chained Assignments
- Assignment and Initialization
- Initialization with Copy Constructor
- Copy Constructors
- Passing Objects
- Returning Objects
- Passing References to Objects

Module 5 : Class Features

- Static Members
- Accessing Static Data Members
- Initialization of Static Data Members
- Static Member Functions
- Friends
- Friend Functions
- Iterator Class
- Arrays of Objects
- Initializing Object Arrays
- Free Store and Class Arrays
- _set_new_handler Function
- Overloading new and delete
- Class Specific new and delete

Module 6 : Inheritance and Polymorphism

- Inheritance
- Type Relations in C and C++
- Derived Classes in C++
- Class Hierarchy
- Redefining Member Functions
- Calling Redefined Members
- Derived Class Constructors
- Base – Derived Class Conversion
- Pointer Conversions
- Virtual Functions
- Polymorphism
- Dynamic Binding
- Virtual Function Implementation
- Virtual Function Table
- Pure Virtual Functions
- Abstract Classes
- Multiple Inheritance
- Duplication of Data Members
- Virtual Derivation

Module 7 : Operator Overloading

- Operator Overloading
- Overloading for Numeric Types
- Overloading Rules
- Overloading Restrictions
- Not Overloadable Operators
- When not to Overload
- Numeric Class Overloading
- Operators as Friend
- Unary Overloading Operator

Module 8 : Templates

- Function Templates
- Macro's versus Templates
- Using Template Functions
- Template Function Overriding
- Class Templates
- Template Array Class
- Template Class Definition
- Template Class Instantiation
- Output Operator as Friend
- Template Class Specialization
- Template Class Static Members
- new and delete in Template Classes
- Template Initialization Statics
- Memory Allocation Statics
- Standard Template Library (STL)

Module 9 : Exception Handling

- Exception Handling in C++
- Memory Exhaustion Handling
- Exception Handling Principles
- Throwing Exceptions
- try Block
- catch Handlers
- Division By Zero Example
- Benefits Exception Handling
- Multiple catch Handlers
- Template Array Class
- Exceptions Array Class
- Matching
- catch Order
- throw List