

C++ Unit Testing

Doelgroep Cursus C++ Unit Testing

De cursus C++ Unit Testing is bedoeld voor developers die **C++** unit tests willen schrijven met het C++ Unit Test Framework Google Test en Google Mock for C++.

Voorkennis Cursus C++ Unit Testing

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

Uitvoering Training C++ Unit Testing

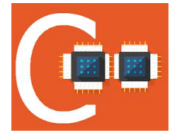
De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met oefeningen. Illustratieve demo's worden gebruikt om de behandelde begrippen te verduidelijken. De tijden van de cursus zijn van 9.30 tot 16.30.

Certificering cursus C++ Unit Testing

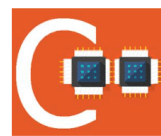
De deelnemers krijgen na het goed doorlopen van de training een officieel certificaat C++ Unit Testing.

Duur: 2 dagen

Prijs: € 1499

[Open Rooster](#)

C++ Unit Testing



Inhoud Cursus C++ Unit Testing

In de cursus C++ Unit Testing leren de deelnemers het Google Test Framework for C++ en Google Mock for C++ te gebruiken voor C++ Unit Testing. Google Test en Google Mock behoren tot de meest gebruikte frameworks voor Unit Testing, maar desgewenst kan de cursus ook worden verzorgd in een ander test framework zoals Boost.Test, Catch2 of Doctest.

Intro Unit Testing

De cursus C++ Unit Testing gaat van start met een introductie in Unit Testing dat erop is gericht individuele units in de source code afzonderlijk te testen. Een unit is hierbij het kleinste onderdeel van de code dat in isolatie kan worden getest zoals een functie of een method van een class.

Test Cases

Vervolgens wordt na een overzicht van andere vormen van testing, uitgebreid aandacht besteed het Google Test Framework. Hierbij wordt de structuur en het opzetten van Test Cases en Fixtures in Google Test besproken.

Assertions

Ook de diverse types assertions komen aan de orde met explicit success of failure afhandeling. Bijzondere aandacht is er hierbij voor exception assertions, floating point comparisons en het gebruik van reguliere expressies.

Test Parameters

Dan wordt ingegaan op het maken van geparametriseerde tests, parameter doorgifte en type-parameterized tests. En ook test events en event listeners worden behandeld.

Test Driven Development

Eveneens onderdeel van het programma van de cursus C++ Unit Testing is de methodologie van Test Driven Development (TDD). De drie regels van TDD en de stappen in TDD worden uitgelegd en de voordelen en de beperkingen van TDD komen ter sprake.

Mocks en Stubs

Tenslotte wordt in de cursus C++ Unit Testing het gebruik van stubs en mocks behandeld. Hierbij komt het Google Mock Framework aan de orde.

Modules Cursus C++ Unit Testing

Module 1 : Unit Testing Intro	Module 2 : Google Test	Module 3 : Assertions
What is Unit Testing? Benefits of Unit Testing Manual Testing Automated Testing Time to Test Unit Test Example Best Practices Other Types of Testing Continuous Integration Regression Testing Usability Testing	Basic Concepts Test Cases Assertions Failures Basic Assertions Binary Comparison String Comparison Simple Tests Test Fixtures Invoking the Tests Set-Up and Tear-Down	Explicit Success and Failure Exception Assertions Predicate Assertions Predicate-Formatters Floating-Point Comparison Type Assertions Regular Expression Syntax Using Assertions in Sub-routines Adding Traces to Assertions Propagating Fatal Failures Asserting on Subroutines
Module 4 : Parameterized Tests	Module 5 : Test Driven Development	Module 6 : Google Mock
Value Parameterized Tests Reading Input Data Passing Parameters General Syntax Typed Tests Type-Parameterized Tests Test Events Event Listeners Using Event Listeners Disabling Tests Enabling Disabled	Traditional Testing versus TDD Three Rules of TDD Steps in TDD Test Cycles Benefits of TDD Limitations of TDD Testing versus Design TDD Adaptation Behavior Driven Development Designing for Testing Code Kata's	Mock Objects Collaborating Objects Mock Implementation Test using Mock Anti Patterns Writing Mock Classes Using Mocks in Tests Setting Expectations Matchers and Cardinalities Using Multiple Expectations Ordered vs Unordered Calls