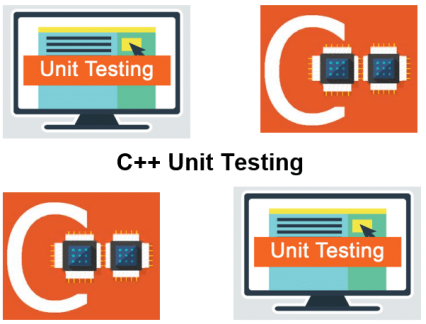
C++ Unit Testing

Duration: 2 days Price: € 1499

Open Schedule



C++ Unit Testing

Audience Course C++ Unit Testing

The course C++ Unit Testing is intended for developers who want to use C++ to write unit tests with the C++ Unit Test Framework Google Test and Google Mock for C++

Prerequisites Course C++ Unit Testing

To participate in this course knowledge of and experience with programming in C++ is required.

Realization Training C++ Unit Testing

The theory is discussed on the basis of presentation slides and is interchanged with exercises. Illustrative demos are used to clarify the concepts discussed. The course times are from 9.30 am to 16.30 pm.

Certification course C++ Unit Testing

After successful completion of the training participants receive an official certificate C++ Unit Testing.

Content Course C++ Unit Testing

In the course C++ Unit Testing, participants learn to use the Google Test Framework for C++ and Google Mock for C++ for C++ Unit Testing. Google Test and Google Mock are among the most widely used frameworks for Unit Testing, but if desired, the course can also be provided in another test framework such as Boost.Test, Catch2 or Doctest.

Intro Unit Testing

The course C++ Unit Testing starts with an introduction to Unit Testing that focusses on testing individual units in the source code. A unit is the smallest part of the code that can be tested in isolation, such as a function or a method of a class.

Test Cases

Subsequently after an overview of other forms of testing, extensive attention is paid to the Google Test Framework. The structure and setup of Test Cases and Fixtures in Google Test is discussed.

Assertions

The various types of assertions are also treated with explicit success or failure handling. Special attention is paid to exception assertions, floating point comparisons and the use of regular expressions.

Test Parameters

Then the course C++ Unit Testing will cover creating parameterized tests, parameter passing and type-parameterized tests. Test events and event listeners are also discussed.

Test Driven Development

Also part of the program of the course C++ Unit Testing is the methodology of Test Driven Development (TDD). The three rules of TDD and the steps in TDD are explained and the benefits and limitations of TDD are discussed.

Mocks en Stubs

Finally the course C++ Unit Testing ends with the treatment of the use of stubs and mocks. The use of the Google Mock Framework is explained here.

Modules Course 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