Do you know why coding guidelines for modern C++ language are needed and required by the safety-related and cybersecurity standards?

Do you know how to effectively and efficiently adopt and comply with the MISRA C++:2023 and AUTOSAR C++14, for new or existing code?

Do you know which coding guidelines to apply the best, or whether to apply multiple coding guidelines?

Do you need to migrate your code to modern C++?

Do you wonder how to address existing non-safety-related guidelines like C++ Core Guidelines?

DE0802 Software implementation in modern C++, in accordance with MISRA C++:23 and AUTOSAR C++

This training introduces the C++ coding guidelines for the implementation of reliable safety-related, and security-related software, but it applies also for non-safety-related software. It introduces multiple C++ coding guidelines but focuses on AUTOSAR C++14 Coding Guidelines and AUTOSAR C++:2023.

Examples, exercises, interactive quizzes, and a hands-on workshop support learning success and understanding of C++-language-specific concepts and problems and how to avoid them.

The training also includes the exida recommendations for adopting coding guidelines in your software development process.

DE0802 Software implementation in modern C++, in accordance with MISRA C++:23 and AUTOSAR C++

Agenda and Content

- Background to coding guidelines
  - What are coding guidelines and why do we need them?
    - C++ language complexity
    - Allowed/prohibited features
- Overview of Modern C++ coding guidelines on the market
  - Coverage of all leading guidelines available
- AUTOSAR C++14 Coding Guidelines – overview
- Traceability of AUTOSAR C++14 to existing standards
  - ISO26262, IEC 61508, ISO 21434, and other safety- or security-related standards
  - Other coding guidelines
- Overview of MISRA C++2023
- Overview of the AUTOSAR C++14 document
  - Guidelines classifications, categories, decidability
  - Guidelines format
- Overview of the AUTOSAR C++14 rules
  - Document structure inspired by ISO C++
  - Rules walk-through and quizzes
  - Introduction to checked and unchecked exceptions concept
  - Introduction to dynamic memory management (6.18.5) rules
- Comparison of MISRA C++2023 and AUTOSAR C++14
  - Advangages / disadvantages of both stanards
  - How to combine both standards

DE0802 Software implementation in modern C++, in accordance with MISRA C++:23 and AUTOSAR C++

- Adopting AUTOSAR C++ in the software development process
  - Process activities
  - Checkers
  - Tools management, configuration, and validation
  - Deviation procedure
  - exida recommendations

Who should attend?

- Software Development Engineers that implement software in C and/or are responsible for the deployment of coding guidelines
- Software Architects

Duration: 2 days à 4 hours (1 day à 8 hours if requested)

Language: English

The training material will be in English.

Location: online

Certificate: Each participant gets a letter of attendance.
After the end of the training, there is a possibility to take the exam including certificate.

For more information, please contact:

Kerstin Tietel
+49 89 44118232
kerstin.tietel@exida.com