



## Embedded and Real-Time Programming Languages

These courses are designed for developers with a basic understanding of programming concepts and are suitable for a wide range of applications, including the development of real-time systems, firmware, and drivers.

The C, C++ languages, and OpenCL for embedded systems category includes courses on language fundamentals, advanced programming techniques, and the use of these technologies in specific embedded systems applications. **ol 2. C Language for Embedded MCUs** The course covers topics such as language fundamentals, advanced programming techniques, and the use of C in specific embedded systems applications. It is suitable for developers with a basic understanding of programming concepts and is designed to provide a strong foundation in C programming for embedded systems.

Upon completion of the course, attendees will be able to develop reliable and efficient software for microcontroller-based systems using C with confidence. **24 h Inquiry** **ol 3. Embedded C++ Programming** The course covers topics such as language fundamentals, the use of C++ templates in embedded systems, advanced aspects such as polymorphism and inheritance, dynamic memory allocation in embedded applications, and the management of C++ exceptions for secure embedded applications.

Additionally, students will learn how to use C++ objects to handle serial transmission/reception of character strings. **18 h Inquiry** **ol 9. OpenCL High Performance Computing (HPC)** is more and more frequent in embedded systems, for graphics rendering, virtual reality or parallel computing. The OpenCL language allows to program in a more or less hardware-independent way complex parallel algorithms that will be able to run on various platforms. **19 h Inquiry** **ol 10. Classic and Modern C++ for Embedded Systems** This course is the combination of the **ol 3 Embedded C++ Programming** course and **ol 10 Embedded Modern C++ Programming** classic and modern C++ programming for embedded systems. **30 h Inquiry**