אקדמיית הקהילה Embedded Makers
בשיתוף מכללת John Bryce מציגות:

C++ for Embedded & Real Time Systems

Online Course | 40 Hours

המבחן של המדינה קבע:

ג'ון ברייס היא המכללה שמשלבת
הכי הרבה בוגרים בעולם ההייטק

Overview

This course introduces the C++ language for use on Real Time and embedded applications. The course covers object oriented programming and the C++ language, highlighting areas of concern for Real Time and embedded development. The course also covers the application of C++ to Real Time systems including interrupt handling, memory issues, and performance issues. The course will allow students to avoid dangerous performance and correctness problems unknown to most C++ developers, including many experienced programmers. During labs, students will solve exercises by writing programs that will illustrate the covered principles. Approximately 40% of the course is practical work.

On Completion, Delegates will be able to

- The core C++ syntax and semantics.
- Object Oriented Advantages, and Principles
- How to write safe, efficient C++ code
- Memory and performance issues associated with C++
- How to access memory & program interrupts in C++

Who Should Attend

The course is designed for Real Time engineers who are embarking on a project using C++ for the first time. It is also targeted at developers currently reluctant to move to C++ from C as they believe it poses too great an overhead. This course will clearly demonstrate both the strengths and weaknesses of C++ versus C.

Prerequisites

Delegates should have a working knowledge of C, and some knowledge of Embedded/Real Time programming.

A Course Introduction
- Course Prerequisites
- Course Objectives
- Course Delivery
- Course Practical
- Course Structure

An Overview of OO Programming and C++
- Review of OOP principles
- Behavior, state, identity, inheritance, polymorphism, abstraction
- History and evolution of C++
- Key features of C++
- C++ as a better and safer C, C++ vs. C, C++ in Real Time systems

The Class Approach
- Grouping of data and functionality
- Syntax of a class declaration
- Syntax of use
- Public and private
- Abstract Data Types
- Program structure

Providing Class Functionality
- Member functions
- Function overloading
- Default arguments
- Ambiguities
- Anonymous arguments
- Resolving scope conflicts
- The Scope resolution operator
- The this pointer

Object birth and death
- Life of an object
- Constructors
- operator new
- Death of an object
- Destructors
- operator delete
- Dynamic arrays

Efficiency, Integrity and Performance Issues
- Enumerations
- Const declarations
- Const member functions
- Const member data
- Inline function mechanism
- Reference variables
- Composite Classes
- An opportunity for reuse
- Embedded / Real Time considerations

Scoping and initialization
- Order of construction
- Member Initialization lists
- Use of fundamental classes

Associative Classes
- Delegating class functionality
- Dynamic associations
- Custody and lifetime
- Constant associations

Operator Overloading
- Operator functions
- Unary operators
- Binary operators
- Global operators
- Member operators
- Subscript operators
- Input operators
- Output operators
- Guidelines
- Embedded / Real Time considerations

Class Properties
- Static data members
- Static member functions
- Nested types
- Forward declarations
- Friend classes

Templates and Container Classes
- Organizing collections of objects
- Template classes
- vector
- list
- Iterators
- Template functions
- Algorithms
- Using the Standard Library
- Embedded / Real Time considerations

Copying and Conversions
- The copy assignment operator
- Copy constructors
- Conversions to a class object
- Conversions from a class object
- Embedded / Real Time considerations

Inheritance
- Extension of existing classes
- Notation, syntax, terminology
- Protected members
- Scoping and initialization
- Multiple inheritance
- Abstract base classes
- Guidelines

Polymorphism
- Modified class behavior
- Virtual functions
- virtual destructors
- Late binding
- Inside the virtual function mechanism
- Pure virtual functions
- Use of pointers to base type
- Guidelines
- Real Time considerations

Embedded and Real Time C++ Considerations

- Comparing C and C++ performance, Performance analysis
- C++ code translated to C
- Inheritance in C
- The Embedded C++ Language Standard
- Program Size Comparisons
- Problems with Exceptions, RTTI, mutable
- Problems with Templates, Multiple Inheritance, Operator Overloading
- Compiling Embedded C++
- Making Objects ROMable
- Encapsulating a ROMable class
- Placing objects at a specific address
- Interrupts and interrupt vectors in C++
- Combining C and C++ code

ג'ון ברייס הדרכה

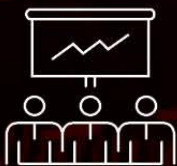מגוון הכשרות טכנולוגיות באמצעות חוויית למידה מתקדמת וחדשנית

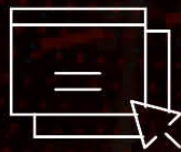סגל מרצים מובילים המורכב ממיטב אנשי המקצוע בתעשייה

קורסים מקוונים בעברית ללמידה עצמית - JB Online Academy

למידה פרונטאלית בכיתות הדרכה משולבת עם לימודי Online

למידה גמישה ודינמית עם כלים מתקדמים בשילוב סימולציות

תכנים עדכניים המותאמים להתפתחות הטכנולוגית בשוק ולביקוש בתעשיית ההייטק

מרכז בחינות והסמכות בינלאומי

מרכז הדרכה מוסמך של החברות המובילות