HackerU

C++ for C Programmers

BT120

40
Academic Hours

C++ for C Programmers

Outline

C++ is undoubtedly one of the most popular programming languages for software development. It brings language enhancements and object-oriented programming support to C. However, C++ is a large and sometimes difficult language, and even with a C background, a programmer needs to understand C++ programming style as well as C++ constructs to get the best out of itl. For experienced C programmers, the course will illustrate how to get the benefits of good software engineering and code reuse by using standard C++ and object-oriented programming techniques in real-world programming situations. This is a hand on course with a mix of tuition and practical sessions for each technical chapter which reinforce the C++ syntax and object-oriented programming techniques covered in the course.

Target Audience

C Programmers wishing to learn or improve in C++

Prerequisites

- Delegates should have a working knowledge of C, and some knowledge of Embedded/Real Time programming.
- Delegates must have solid experience of C including structures (i.e. struct and/or class); declaration and use of pointers; function declaration, definition and use with call by value or call by pointer; dynamic memory allocation (i.e. malloc and free, or new and delete); multiple source file projects (requiring project files or makes files)

Objectives

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++

Content

Module 01
A Course Introduction

- Course Prerequisites
- Course Objectives
- Course Delivery
- Course Practical
- Course Structure

Module 02
An Overview of OO Programming & 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

Module 03
UML Brief overview

- General overview on UML
- Class Diagram
- Sequence Diagram

Module 04
The Class Approach

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

Module 05
Providing Class Functionality

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

Module 06
Object birth and death

- Life of an object
- Constructors
- operator new
- Death of an object
- Destructors
- operator delete
- Dynamic arrays

Module 07
Efficiency, Integrity & 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

Module 08
Composite Classes

- An opportunity for reuse
- Scoping and initialization
- Order of construction
- Member Initialization lists
- Use of fundamental classes

Module 09
Associative Classes

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

Module 10
Operator Overloading

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

"C++ is undoubtedly one of the **most popular programming languages** for software development"

Module 11
Class Properties

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

Module 12
Templates and Container Classes

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

Module 13
Copying and Conversions

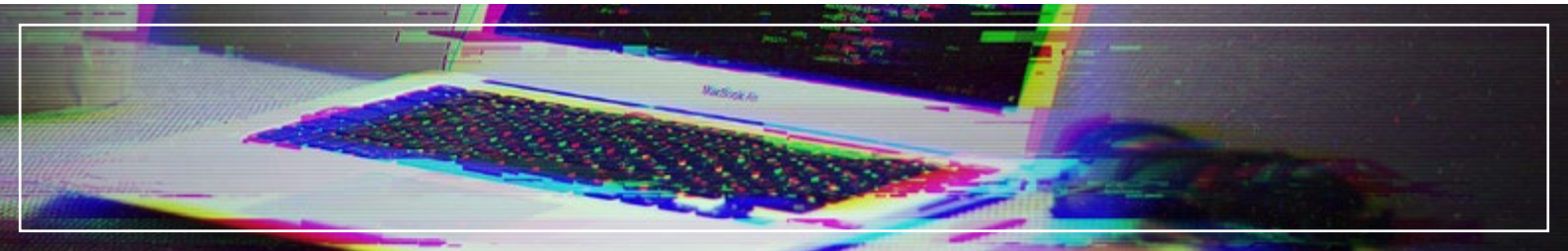
- The copy assignment operator
- Copy constructors
- Conversions to a class object
- Conversions from a class object
- Embedded / Real Time considerations

Module 14
Inheritance

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

Module 15
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



Module 16

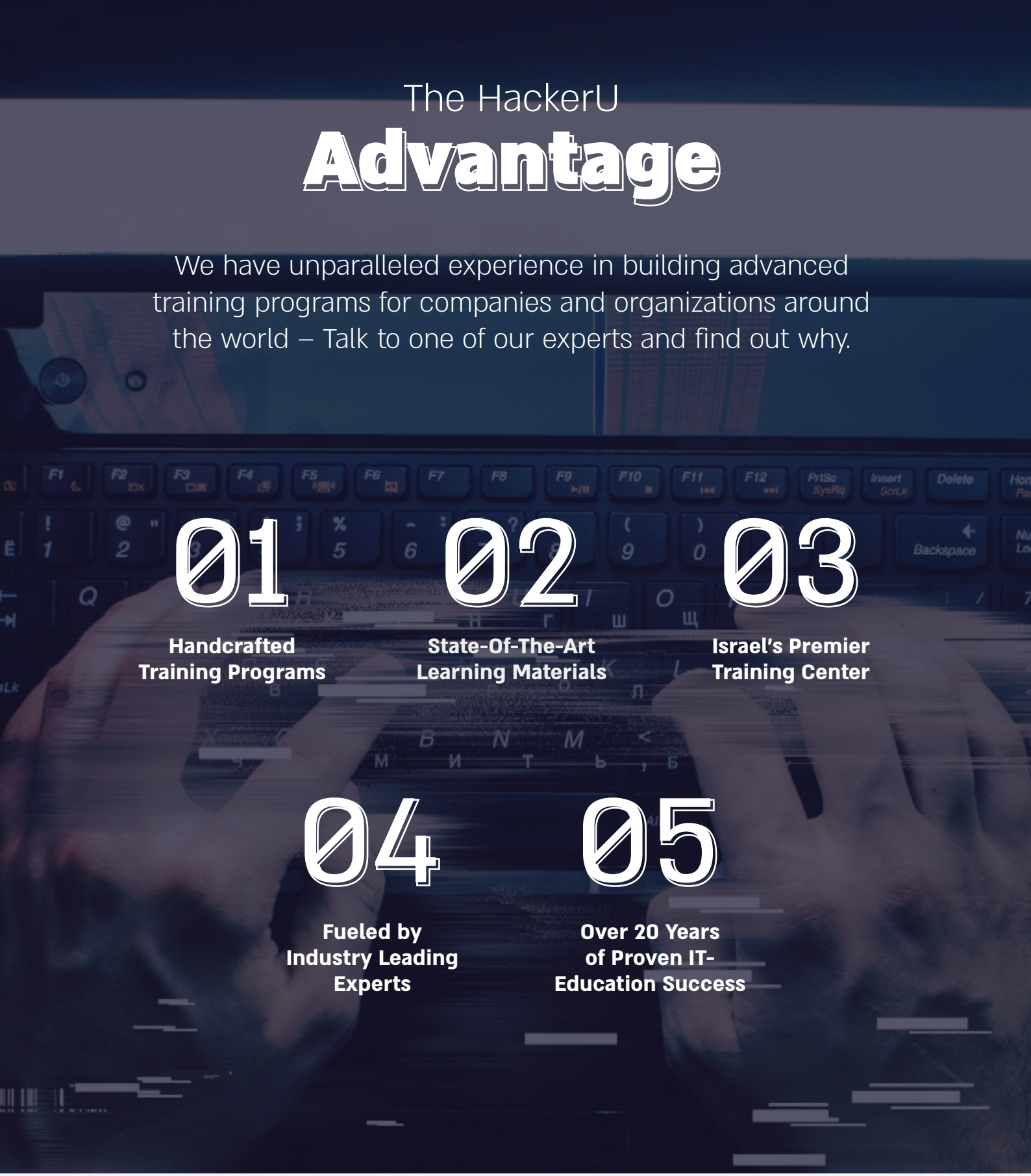
Embedded and Real Time C++ Considerations

- 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

Module 17

OOC (Object Oriented in C)

- Modularity and correct API Definition
- Encapsulation and Information hiding
- Inheritance



The HackerU Advantage

We have unparalleled experience in building advanced training programs for companies and organizations around the world – Talk to one of our experts and find out why.

01

Handcrafted Training Programs

02

State-Of-The-Art Learning Materials

03

Israel's Premier Training Center

04

Fueled by Industry Leading Experts

05

Over 20 Years of Proven IT-Education Success

info@hackerupro.com

www.hackerupro.com