C++ 11, Part 3 of 4: Data Structures and Pointers

page 1

Meet the expert: Peter Thorsteinson has been working for two decades in many areas of Microsoft-based software development technologies, including all the latest languages, frameworks, and tools. He has been involved in several large scale software development projects and has authored and co-authored several books and courses relating to ASP.NET, AJAX, JavaScript, WPF, WCF, WF, ADO.NET, and LINQ. Peter is currently focusing on ASP.NET MVC, jQuery, Task Parallel Library, Windows Azure, and SharePoint Development.

Prerequisites: This course assumes that students have some rudimentary understanding of general programming concepts. Watching Parts 1 and 2 in this series will be helpful. As with any such course, the more experience you bring to the course, the easier it will be to understand the material that is presented. As a result, if you know C# or Java, that might be somewhat helpful.

Runtime: 02:23:37

Course description: The key strengths of C++ are its object-oriented features and its basis on classes and structures. Discover how to define data types, reference types, and value types and also how to implement object members inheritance, enumerations, unions, and typedefs. Learn how pointers work, including dereferencing, passing pointers, and pointer arithmetic to access data in objects. Finally, compare and contrast pointers with smart pointers for memory cleanup.

Course outline:

Classes and Structs
- Introduction
- Classes and Structs
- Demo: Classes and Structs
- Demo: Constructors
- Demo: Inheritance and Scope
- Demo: Complicated Example
- Summary

Enums
- Introduction
- Enums
- Demo: Defining an Enum
- Demo: Using Enums
- Summary

Unions
- Introduction
- Unions
- Demo: Union
- Demo: Discriminated Union
- Summary

Typedefs
- Introduction
- Demo: Simple Typedef
- Demo: Complex Typedef
- Summary

Introduction to Pointers
- Introduction
- Demo: Intro to Pointers
- Demo: Swapping Values
- Summary

Pointers and Array Indexing
- Introduction
- Demo: Address-of and Dereference
- Demo: Arithmetic and Array Indexing
- Summary

Using Const with Pointers
- Introduction
- Demo: Using Const
- Demo: Heap Objects and Member Operator
- Summary

Pointers to String Literals
- Introduction
- Demo: String Literals
- Demo: Voids and Nulls
- Summary

References
- Introduction
- Demo: References vs. Pointers
- Demo: References
- Demo: Risky Referencing
- Summary

Smart Pointers
- Introduction
- Smart Pointers
- Demo: Raw Pointers
- Demo: Unique and Shared Pointers
- Demo: Weak Pointers
- Summary