C++ Windows Programming using MFC	*5 Days*

Course Aims:

This course is designed to provide developers, who already have some C++ experience, with the experience they need to develop windows based C++ applications using the Microsoft Foundation Classes (MFC).

Course Outline:

Overview of Windows Programs:
- The architecture of a non-MFC program
- The winmain() function
- Processing windows messages

Overview of MFC:
- The evolution of MFC
- The MFC Classes
- The MFC Exception Classes
- Architecture of an MFC program
- InitInstance()

Detecting Memory Leaks:
- Memory leakage
- The CMemoryState Class

The App Wizard:
- SDI Applications
- App Wizard output
- MDI Applications
- The Class Wizard
- The Draw application

Application Framework:
- Document / View architecture

Working With Different Types of Project :
- The different types of view class

Menus, Toolbars & Other Resources:
- Creating & editing menus
- Creating & editing toolbars
- Message handlers

Drawing in a Window:
- Windows GDI
- Device Context
- Mapping Modes
- The CDC class
- Using the mouse

MFC Collection Classes:
- CArray
- CList
- CMap
- CTypedPtrList

Dialogs & Controls:
- Dialog resources
- Controls
- Modal & Modeless Dialogs

Serialization & Printing:
- CArchive & CFile
- Serializing the Document class
- The printing process
- Print & Print preview
- Enhancing the printing process
- Printing multiple pages

Database Applications:
- DAO vs ODBC
- Creating a database application
- RFX
- Filtering & Sorting

Introduction to COM & OLE:
- The MFC OLE classes
- OLE server & container applications

MFC ActiveX Controls:
- Creating an ActiveX Control
- The ActiveX Control test container
- Property sheets / property pages

Multithreaded Applications:
- Threading overview
- CWinThread
- Thread Synchronisation

Target Audience:

Systems and applications programmers who will be developing or maintaining C++ Windows programs. Anyone who wants a practical understanding of Windows C++ programs using MFC will benefit from this course.

Assumed Knowledge:

Participants should have a good knowledge of programming techniques in C++. A good knowledge of how to implement the OO principles of encapsulation, inheritance & polymorphism is required.