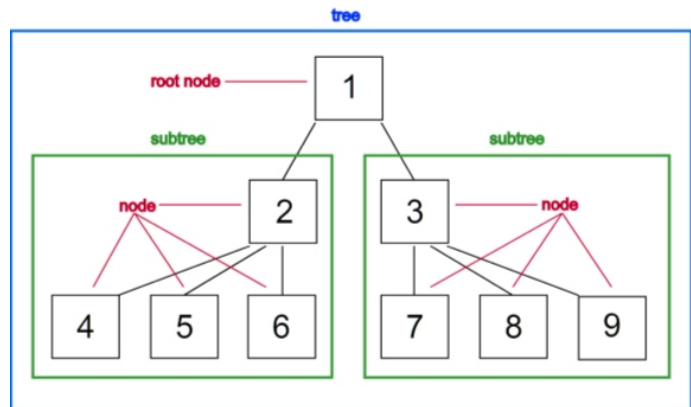


C & C++



Corporate Trainer's Profile

Corporate Trainers are having the experience of 4 to 12 years in development , working with TOP CMM level 5 companies (Project Leader /Project Manager) qualified from NIT/IIT/IIM and work exp in USA and UK.



Capability Maturity Model level Project Standard*** :-

The Capability Maturity Model (CMM) is a method for evaluating the maturity of organizations on a scale of 1 to 5.

Get the Opportunities to work on Client Projects Of US/UK, which follow the all standard of CMM level 5 Company.

Projects



Advanced C++ Programming for Unix and Linux - A 5 Day Course

Synopsis

This course covers the mechanics of building complex Unix / Linux based applications in C++. It is a collection of examples and case studies that are important, yet do not seem to be dealt with elsewhere.

A large part of the course explores techniques for implementing wrappers around Unix API system calls in order to implement functionality at a higher level of abstraction, by hiding much of the low level functionality inside a suitable class.

The course also covers techniques for developing Graphical User Interfaces using Widget sets implemented in C++, such as Qt+ and Gtk+, as well as the Motif widget set (which is implemented in C).

The principles of developing distributed applications using CORBA are introduced, as are C++ object oriented approaches to building Socket based applications.

A major section of the course deals with implementing Posix threads based applications in an object oriented manner.

Suitable for

The course is for experienced C++ and Unix programmers, who would like to take a more object oriented approach to Unix application development.

Attendees are expected to have an engineering "mind set" - i.e. prepared to "get their hands dirty" by experimenting with ideas and programming approaches.

The course is also suited to those who will be designing and implementing large complex application frameworks that will be running on Unix / Linux platforms.

Prerequisites

A good understanding in C++ programming

Several months experience in practical C++ programming

Delegates must be fluent in C++ and keen to learn about methods of building better software and improving their understanding of what works well in a C++ environment. Delegates who will obtain the most from the course are those who enjoy learning and questioning, who adopt an intelligent and inquisitive approach to their task.

Course Skills

C++ Graphical User Interface Implementation

Distributed application development using CORBA and C++

Implementing wrappers around C based API functionality

Working with threads in an object oriented way

Working with virtual classes and virtual inheritance

Understanding object oriented approaches to network programming and network management

Implementing data persistence using serialisation techniques, relational database techniques and object oriented database techniques

Other Information

The course contains a number of challenging exercises. For many of the exercises a lot of the

supporting code will be provided, so that students can concentrate on applying one or more key concepts.

Exercises will consist of a core problem, followed by several challenges of increasing difficulty. The challenges are meant for post course skills development.

- Implementing a graph search algorithm using the STL
- Setting up a CORBA based information service
- Developing a Motif based query form in C++
- Implementing Tetris in C++ using Qt+ or Gtk+ as the basis for the user interface
- Implementing a socket based video data streaming service in C++
- Devising a multi-threaded simulation of a manufacturing plant
- Exploring and improving the functionality provided by a basic serialisation library

Contents:

- An overview of C++ - the language, patterns and idioms
- Inheritance, Polymorphism and all that
- Function Objects
- Templates and the Standard Template Library
- Mixed C and C++ programming
- Key patterns - Singleton, Observer, Visitor, Composite
- Templates and the Standard Template Library
- Namespaces and scope
- An overview Unix technologies
- The POSIX API and its variants
- Inter-process communication under Unix/Linux
- The Berkeley Sockets API
- POSIX Threads (p-threads)
- The Unix file system
- X windows
- Techniques for Devising and Implementing Wrapper Classes
- Operating system resources as objects
- The create a resource, get a handle, manipulate the resource via system and library calls that take a handle as an argument API model
- Creating classes corresponding to operating system resources - that contain the handle as a member variable, and methods for manipulating the resource that invoke the underlying system and library calls
- Wrapping up Motif Widgets in C++ - a case study
- Extending functionality beyond wrapper classes
- Managing a collection of child processes
- Devising two way communication with objects that contain a pair of pipes
- Creating shared memory objects with built in critical sections
- Implementing a Model-View-Controller pattern using Motif Widgets Working with P-threads the object oriented way
- Designing and implementing a p-thread class
- Devising and implementing a thread local storage scheme
- Managing a pool of worker threads
- Inter-task communication mechanisms for use with p-threads
- Working with C++ based widget sets

- Commonly occurring widgets - buttons, menus, dialogs, list boxes ...
- Layout managers
- Overview of Qt+
- Overview of Gtk+
- Deriving new widget classes
- CORBA and C++
- Overview of CORBA
- MICO - Corba implementation
- ORBACUS - Corba implementation
- Implementing Client / Server applications in C++ and CORBA
- Data Persistence
- Serialisation - simple and complex
- Serialisation libraries
- Use of Smart pointers
- Persistence using relational databases
- Persistence using object relational databases
- Persistence using object oriented databases
- Network programming
- Object oriented perspective on TCP/IP
- Wrapping up the Sockets API in C++
- Building application services derived from a Socket class
- Network management with SNMP
- An overview of SNMP++