Outline:

Phase I: Graduate Review

The first phase of the Workshop focuses on aligning the participants' skill sets and understanding of software fundamentals with what should be expected from a University graduate. Special attention is paid to correcting common academic misconceptions, as well as highlighting important issues which may have been glossed over during their academic studies.

- Linux from Scratch
  - o What Developers Need to Know About System Administration
  - o User Environment
  - o Hacking The Shell
  - o Development Environment & Tools
- Enough SQL to Survive
- Enough ASM to Survive
- Advanced In-depth C
  - o C Traps & Pitfalls
  - o Optimization & Portability Issues
- Making the Most of Data Structures
- Practical Guide to Operating System Fundamentals
  - o Memory Management
  - o Process Management
  - o Process Structure
  - o IPC (Message Queues, Shared Memory, etc)
  - o Multi-Threading & Synchronization
- The Build Process & its Products
  - o Debugging the Preprocessor, Linker, Symbol Tables, etc
  - o Shared Objects/DLLs & Libraries
  - o Debugging The Build Process
- Network Programming
  - o TCP/UDP/IP
  - o RS-232/485
  - o Single-Threaded Servers
  - o Buffering I/O
  - o Protocol Design & Development
- Advanced C++
  - o C++ Internals, Optimizations & Language History
  - o C++ Traps & Pitfalls
  - o Advanced Polymorphic Strategies
  - o STL
  - o Multi-Paradigm Language Features

- Object Oriented Programming
  o Design Patterns (GOF)
  o Idioms (late evaluation, letter-envelope, v-ctor etc.)
  o Generic Programming Techniques
  o Interface Design

Phase I: Basic Training

- The C Programming language
- C++ & OO Programming
- Data Structures & Algortihms
- Computer Architecture
- Operating Systems Theory
- Software Engineering

Phase II: Advanced Programming

- Advanced In-depth C
  o C Traps & Pitfalls
  o Optimization Issues
- Advanced Recursion & Data Structures
- Operating System Fundamentals
  o Memory Management
  o Process Management
  o Process Structure
  o IPC (Message Queues, Shared Memory, etc)
  o Multi-Threading & Synchronization
  o Shared Objects
- The Build Process & its Products
  o Preprocessor, Linker, Symbol Tables, libs, etc
  o DLLs & Libraries
- Network Programming (TCP/IP)
- Object Oriented Programming & Advanced C++
  o C++ Internals
  o C++ Traps & Pitfalls
  o Advanced Polymorphism
  o Design Patterns (GOF)
  o Idioms (late evaluation, letter-envelope, v-ctor etc.)
  o Generic Programming Techniques
  o Interface Design
  o STL

Phase III: The Real World

- Traditional Project Workflows
  o The Development Process
  o Waterfall, Spiral, Fast Prototypes, etc
  o Development Process By-Products (documentation)
- Advanced Debugging Techniques

- - Memory Overruns
  - Memory Leaks
  - Stack Structure
  - Enough Assembly to Survive
  - Debug vs release Mode
  - Debugging Multithreaded Systems
- Accurate Task Estimation
- Revision Control
- Code Reviews
- Brain Storming as part of the Development Process
- Working as Part of a Team

Phase IV: Commando Training

- Fundamentals of Embedded/RT
- Multithreaded Programming Traps & Pitfalls
- Network Programming Traps & Pitfalls
- Agile Processes (Scrum)
- Design Patterns & Frameworks for Embedded/Real-Time
- Design Patterns for Multi-Threaded Systems
- Advanced Templates
- Framework Design
- Multi-Paradigm Techniques
- Advanced STL
- Plug & Play
- Interface Design
- Embedded Targets (Windows CE x86)
- MFC
- Anti-Patterns
- Presentation of self-study subjects

A Two-Tier Mission which employs:

- Multi-threaded Programming
- Communication issues
- Using Design Patterns
- Multi-Platform Development
- Plug-and-Play
- Taking Advantage of STL
- Generic Programming
- Building Frameworks
- Self-Study & Group work.

* Additional subjects (Java, C#/.Net, Linux, VxWorks etc) as per customer requirements