Advanced C Programming

Course Summary

Description

This course is intended to expose the intermediate level C Programmer to advanced concepts. Concepts include advanced pointer usage, dynamic memory allocation and deallocation, advanced input/output, exception handling, advanced techniques and pitfall avoidance. The class may be taught on Windows based C-development studios or any UNIX variant.

Topics

- Pointers
- Memory / Dynamic Memory
- I/O & Function Topics
- Debugging / Exception Handling
- Techniques

Audience

This class is intended for C programmers needing to extend their skill sets to more advanced concepts.

Prerequisites

The student should have a thorough understanding of C syntax, the standard C libraries, C programming constructs and C development tools. C experience of 6 to 12 months is suggested. (If taught in a Unix environment, student must have a good grasp of Unix concepts and be familiar with Unix text editing.)

Duration

Five days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Advanced C Programming

Course Summary

I. Pointers
- A. Review of Pointer Basics
- B. Pointers to Data Structures
- C. Pointers to Functions
- D. Pointers to Pointers
- E. Pointers to Multi-Dimensional Arrays
- F. Pointers to an Array of Pointers

II. Memory / Dynamic Memory
- A. Review of Dynamic Memory
- B. Linked Lists
- C. Binary Trees
- D. Stacks / Queues

III. I/O & Function Topics
- A. Binary I/O
- B. Problems with String Copy Functions
- C. Problems with Standard I/O
- D. Missing Prototypes
- E. Varying Number of Function Arguments
- F. Data Hiding
- G. Passing and Returning Composite Data Types

IV. Debugging / Exception Handling
- A. Exception Handling Techniques - assert()
- B. Signals and Long Jump
- C. Benchmarking Code
- D. Debugging Techniques / Functions

V. Techniques
- A. ANSI vs. K & R standards
- B. Portability
- C. Efficiency Topics
- D. Design Topics
- E. Wrap-Up Workshop

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically