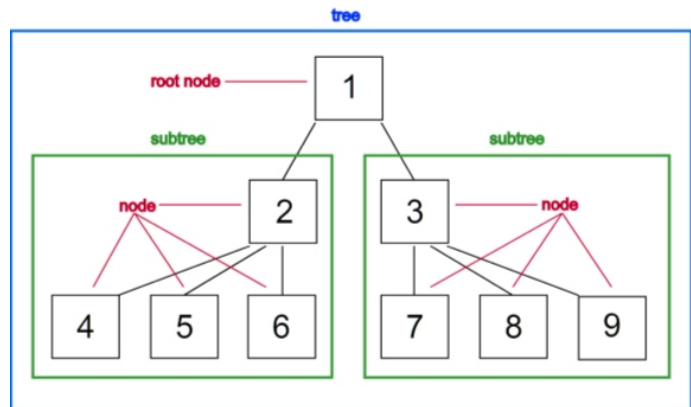


C & C++



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



Introduction to C Programming for Embedded Systems

5-day Course

synopsis

This course discusses C programming in the context of implementing C applications for 8 bit and 16 bit micro-controller architectures. A distinction is made between pure ANSI C programming and use of pragmas and extensions as found with various embedded C compilers targeted at specific microcontrollers. General ANSI C programming is taught using a PC oriented IDE such as Microsoft's Visual Studio or the GCC compiler under Eclipse. Embedded target specific aspects are illustrated using one or more C compilers for PIC16, PIC18 and PIC24 / dsPIC architectures.

You will also learn basic techniques for mixed C and Assembler programming. The hands on exercises also cover the use of simulation, debugging and In Circuit Debugging techniques, as well as techniques for In System Programming.

Course Objectives

Students successfully completing this course will have a sound understanding of C programming. You will understand the basic C data types, arrays and pointers. You will have a good appreciation of data structures and their uses, and the use of pointers to data structures and arrays of data structures. You'll learn how to use arrays to implement circular buffers and how to use them, and how to use arrays to implement stacks, and the uses of software stacks. You will also be introduced to basic techniques of memory management and programming with dynamic data structures - being able to implement circular buffers and stacks using singly linked and doubly linked lists. You'll also learn techniques for implementing interrupt handler code in C, multi-module programming including applications containing a mixture of C and assembly language modules, and techniques for manipulating hardware registers and special function registers.

Suitable for

Programmers and engineers who already have some understanding of programming and who now wish to gain a solid understanding of the use of C for embedded systems software development.

Prerequisites

Attendees should have a basic knowledge of programming and computers. No previous knowledge of the C language is assumed.

Delivery

This is instructor led C training. Each section of the material covered by the tutor is followed by hands-on practical exercises for which worked examples of the solutions are typically provided.

Scheduled and On-site Courses

Courses in this subject are scheduled on an 'ad-hoc' basis. We can arrange a course at our Carshalton centre or on customer site for any client wishing to send two or more

Contents

- An Introduction to C
- Basic data types
- Integer types
- Floating point types
- Pointers (pointer arithmetic, pointer dereferencing)
- Arrays
- The relationship between arrays and pointers
- Enums
- Programming constructs
 - if - else statements
 - switch statements
 - while loops and do - while loops
 - for loops
 - goto statements
- Functions in C
- Function prototypes
- Understanding C's call by value mechanism for passing arguments
- Implementing functions
- Calling functions
- Using pointers to pass values by reference
- Passing arrays to functions
- Understanding C's return value mechanism
- Using pointers to return values by reference

Data Structures

- Defining data structures
- Accessing and manipulating the elements of a data structure
- Pointers to data structures
- The arrow (->) operator
- Using a pointer to traverse an array of data structures
- Advanced topics
- Pointers to pointers
- Function pointers
- Const and static keywords - their meanings and uses
- Casts and casting
- Using void * pointers
- Data structures containing pointers to data structures
- Data structures containing function pointers
- Function pointers and table driven programming
- Bit fields in data structures
- Unions
- Using arrays to implement circular buffers and stacks
- Modules as data structures and interfaces

- Dynamic data structures and their uses
 - o Singly linked lists
 - o Doubly linked lists
- Overview of the C standard library
 - putc, putchar, getc, getchar
 - printf, scanf
- String manipulation functions (strcpy, strcat, ...)
- Embedded System Oriented Topics
 - Setting up port addresses
 - Defining bit oriented data structures
 - Compiler specific features
 - o #pragmas
 - o Non ANSI C extensions
 - o Accessing and manipulating special function registers
 - o Implementing interrupt handlers
 - o Calling assembly code from C
 - o Calling C from assembly code
- Linkers and linker scripts
- Understanding target peripheral device libraries and their uses
- Implementing your own device libraries
- Timers and their uses
 - o Hardware timers
 - o Software timers
- Basics of event driven programming
- Port I/O - flashing LEDs
- Software delays
- Keypad scanning
- Basic RS232
- Basic data acquisition and sensor interfacing
- Working with EEPROM (on chip and SPI/I2C)