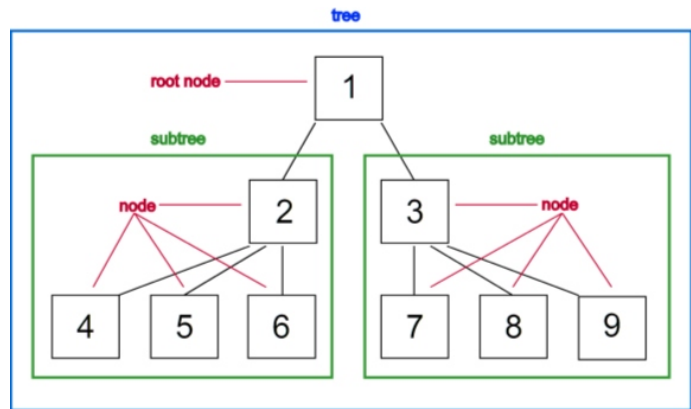


C & C++



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



Combine Modules for Customised C Training

Most of our C course modules are outlined below. They can be combined in different ways to produce a range of specialist C courses. With judicious selection, a purchasing company can build itself a bespoke C training solution from these modules at a fraction of the usual cost. Please ring our training advisors for help in assembling a course with appropriate content, speed of progress, prerequisite knowledge, and consistent subject matter, etc.

N.B. GBdirect write fully-bespoke courses for clients in all of our specialist areas of consultancy. If you have such a requirement and the budget for it, please contact our course developers for advice and support.

Our C Programming Modules

The History of C Programming

The Structure of the C Program:

Compiling a C Program

Variables & Constants:

Variable Names; Data Types; Declarations

Assignment Operator; Constants

Initialisation of Variables; Enumeration Types

printf(); getchar()

Expressions & Operators

Arithmetic Operators; Expression Evaluation

Increment & Decrement Operators

Bitwise Logical Operators

Compound Assignment Operators

Type Conversion; Comma Operator

Order of Precedence

Flow Control of a Program

Types of Flow Control Statements

Logic States; Relational Operators

Boolean Logical Operators; if-else Statement

if-else-if Chain; Nested if Statements

Conditional Operator; Compound Statements

Switch Statement; While loop; Do-while Loop;

For Loop; For Loop Variations

Break & Continue Statements

exit() Statement; goto Statement

Arrays

Theory of Arrays; Range Checking; Initialisation

Unsize Array Initialisation

The Standard I/O Library

getchar() & putchar(); gets() & puts()

printf(); scanf()

Functions

The Definition of a Function; Function Parameters

Non-Returning Functions; Void Functions

The Return Statement
Functions Returning Non-integers
Passing Arrays to Functions; Recursive Functions
Scoping
Storage Class Types; Automatic Variables
Register, Static & External Variables
Initialisation Rules for the Storage Classes
Pointers
The & and * Operators
Pointers & Function Arguments
Pointer Arithmetic; Pointers & Arrays
Arrays as Function Arguments
Pointer Initialisation; Arrays of Pointers
Pointer to Pointers; Command Line Arguments
Pointers to Functions
File Handling
The File Pointer; fopen()
stdin, stdout & stderr File Pointers; fclose()
fgetc(); fputc(); ungetc(); fgets(); fputs()
fprintf(); fscanf(); fread() & fwrite()
fseek(); ftell(); rewind(); fflush(); File Status
Structures & Unions
Members of a Structure Variable
Pointers to Structures
Structure Initialisation & Assignment
Structures & Functions; Nested Structures
Bit Fields; Unions; sizeof(); typedef;
The C Preprocessor
The Pre-processor Operation
#include; Defined Constants
Defined Macros; Macro Side Effects
Deleting a Definition; Conditional Compilation
#ifdef & #ifndef; #line ANSI/ISO C Standard Header Libraries
assert.h, errno.h, math.h
stddef.h, stdlib.h
Character & String Manipulation
stdlib.h, ctype.h, string.h, memory.h,
String Formatting Function
Date & Time Functions
time.h, clock(), time() & ctime()
The tm structure
Memory Allocation
malloc.h, malloc(), free(), calloc(),
realloc(), Dynamic Arrays, Link Lists
System Call I/O
stdio.h, open(), creat(),
Higher Order Open Modes

close (), read () & write (), lseek ()
File Access via both System & Standard I/O Functions
Temporary Files
File Handling
File Access, File Status
Change Access & Modification Times
unmask (), chmod (), chown ()
link () & unlink ()
Directory Handling
Change Working & Root Directory
Making & Removing a Directory
File Tree Walking
UNIX Command Execution
System ()
Piped UNIX Commands
Environment Information
Program Environment Arguments
getenv() & putenv ()
Multiprocessing
Initiating Processes; Parallel Processing
Inter-Process Communication; dup (), fcntl () Process Control
Process Exit & Process Abort
sleep (), nice (), brk () & sbrk (), ulimit ()
Process Identification
Signal Handling
User Information
Real & Effective User & Group IDs
/etc/passwd Information
/etc/group Information
Function Control
Variable Parameters
Inter-function Jumps
The Curses Library
Using the Curses Library
The Basic Curses Routines
Moving About & Writing to the Screen
Reading from the Screen
Input & Output Modes
Screen Editing
Screen Attributes
Creating Windows
The Window Routines
Makefiles
How Make Works; The Makefile
Multiple-file Programs; Make Variables
Suffix Rules; Program Libraries

- Shell Commands
- Command Line Options
- Miscellaneous Features
- The Source Code Control System
- SCCS Tree Structure
- The admin Command; The Get Command
- Ungetting a File; The Delta Command
- Other SCCS Commands
- Running SCCS from a C Program
- The Concurrent Versions System
- CVS Overview
- Creating a repository
- Setting Up Your Repository
- CheckIng Out A Working Directory
- Making changes to files
- Merging changes
- Committing changes
- Adding and deleting files
- Renaming Files
- Writing good log entries
- Handling conflicts
- Identifying Conflicts:
- Resolving Conflicts
- Understanding Conflicts
- Locking