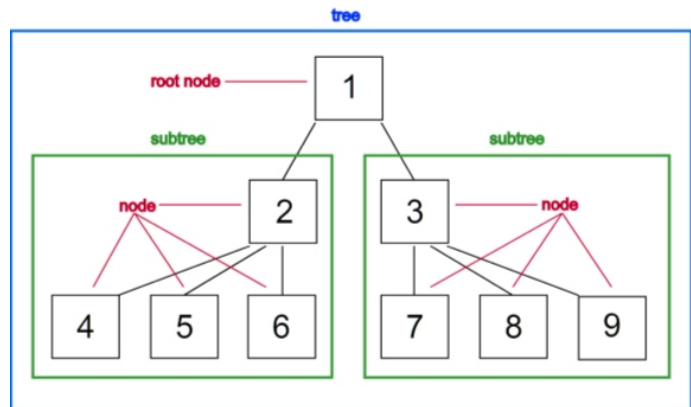


C & C++



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



C++ Programming for C Programmers - A 4 day Course

Synopsis

This in-house course is designed to give a solid grounding in the key aspects of C++ and the Standard Template Library (STL) to those who already have experience of programming in C or a similar language. Where there is no background in C we recommend our Introduction to C++ Programming course instead, as that course also covers the elements of C that are applicable to C++.

The course covers all of the important features of the language, ensuring an understanding of the principles which underpin the design of effective, high-quality delivered software.

The course is based on many years' experience of teaching C++ and avoids the common pitfalls which less mature courses often stumble across. We focus on the needs of day-to-day users of the language who will have to not only use existing libraries of classes but also to understand the mechanisms used in the implementation of those libraries. C++ contains a number of features which, although fascinating from a technical point, are not directly relevant to developing a useful degree of competence with the language. Depending on the likelihood of encountering these, they are either deferred to an advanced course or covered in enough depth to ensure a reading ability only.

The course takes a pragmatic approach to the use of Object Orientation. Examples are carefully chosen to lead students naturally to the concepts of Object Based and Object Oriented programming without being dogmatic about whether these techniques are essential in every case.

Suitable for

Programmers and designers who already have fluency in C (or who have attended the introductory C basics course) and who now wish to gain a solid understanding of the use of C++ for software development.

Prerequisites

Practical experience of software development
Familiarity with procedural programming concepts
Familiarity with the C is vital, but additional knowledge of languages such as Java, Perl, PHP or Javascript also make an excellent starting point.

Course Benefits

The course ensures that delegates understand what is necessary to both design and use C++ classes in the production of high-quality software. The course is designed with programmers in mind and it concentrates on building students' C++ skills as quickly as possible.

Other Information

Delegates receive comprehensive course notes and a reference textbook. The course notes include worked examples to all exercises. The source code to worked examples can be made available on request to GBdirect Ltd.

Contents:

- Background to C++
- User-defined types
- Encapsulation
- Polymorphism
- The Standard Template Library
- Namespaces
- Exercises
- Classes
- Private and public members
- Member functions
- Constructors
- Destructors
- References
- Static Members
- Exercises
- Further Techniques with Functions
- Prototypes
- Linkage
- Overloading
- Ambiguity
- Default Arguments
- Exercises
- Managing Memory
- New and delete
- Object copying
- Copy constructor
- Assignment operator
- The this pointer
- Exercises
- Inheritance
- Simple inheritance
- Polymorphism
- Object slicing
- Base initialisation
- Virtual functions
- Exercises Operator Overloading
- NB: this is a brief introduction
- Operator functions
- I/O operators
- Friends
- Member operators
- Conversions
- Exercises
- Templates
- Introduction
- Template classes

- Declaration and instantiation
- Template functions
- Exercises
- Standard Template Library
- Namespaces reviewed
- Strings
- Vectors
- Iterators
- Hashes
- Iostreams
- Other types
- Exception Handling
 - Purpose
 - Simple use
 - Exceptions and derived classes
 - Function exception declarations
 - Unexpected exceptions
 - Exceptions when handling exceptions
 - Resource capture and release
- Exercise and review
- Worked solutions to all exercises