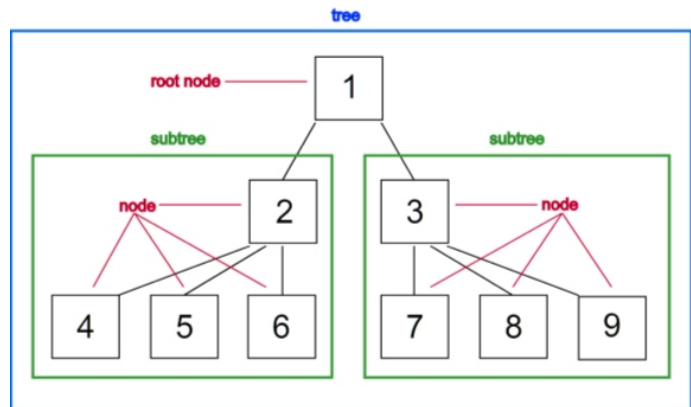


C & C++



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 - A 5 day Course

Synopsis

Unlike many other introductory C++ courses that use Microsoft, or Borland integrated development environments this course is based on the gnu C++ compiler. Programmers attending this course will learn to write carefully thought out and well implemented C++ code and learn a lot about the GNU C++ compiler at the same time. Principles of object oriented analysis and design and of UML are covered. The relationship between C and C++ is explored, and the course includes a thorough overview of C syntax.

Intended Audience

The course is for experienced programmers, with a basic knowledge of C, who will be developing applications in C++, predominantly in a Unix or Linux environment. The compiler used on the course will be the GNU C++ compiler running on a Linux workstation. Knowledge of Object Oriented Analysis and Design, and of UML, although useful is not needed, as these topics will be covered in sufficient detail in the course where appropriate. This course can also be run on Microsoft Windows platforms using the Cygwin port of GNU to Microsoft Windows.

Key Skills

- Understanding the key aspects of object oriented analysis and design and UML
- Use of the GNU C++ compiler
- Handling of Input/Output in C++
- Knowing when to use public and private inheritance
- Working with virtual classes and virtual inheritance
- Understanding templates and essential features of the Standard Template Library (STL)
- Mastering basic patterns and idioms - such as the handle-body idiom, the singleton pattern, the composite pattern and the observer pattern
- Implement simple Graphical User Interfaces using the Qt+ widget set
- Appreciate the issues involved in implementing data persistence

Practical Work

The course contains many simple programming exercises to reinforce the theory, and to stimulate understanding. Coverage of make and makefiles, and the various GNU C++ compiler options will be included at appropriate points in the course.

- Creating and manipulating objects and displaying their attributes
- Experimenting with public and private inheritance
- Experimenting with virtual classes
- Implementing copy constructors
- Exploring deep vs. shallow copying techniques
- Understanding reference counting and the copy on modify idiom
- Implement an instance of the handle-body idiom
- Explore some uses of the Singleton pattern
- Build a simple graphical user interface
- Demonstrate the Observer pattern in action

Implement a templatised singly linked list
Writing and reading a table of records to and from a file
Saving a collection of objects and retrieving a collection of objects in binary format to and from a file
Appreciate some of the functionality provided by the Standard Template Library
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 delegates on the same course.

Contents:

A background to object oriented thinking
Objects as things having attributes and methods
The message passing model of object interaction
Objects as instances of some class
Relationships between objects
Hierarchies of object classes
UML notation for describing classes and class hierarchies
UML for describing relationships between objects
Building more complex objects - composition and aggregation
Practical applications leading to object oriented approaches
Structured programming - modularity and information hiding
Simulation and modeling of complex communications, manufacturing and logistics systems
Artificial Intelligence (AI) - problem solving, planning, searching problems
Graphical User Interfaces - window managers, widgets and controls
Brief History of C++
Early C++ (translation of C++ code to C code)
C++ is not a pure object oriented language - C++ as an extension of C
C++ as an evolving language and the development of the ANSI C++ standard
An overview of the C in C++
Base data types (numeric char, integer and floating point types)
Pointers and arrays
Function prototypes and function pointers
Structs, pointers to structs and dynamic data structures
Typedefs, enums and macro
I/O in C
Replacing C based I/O (printf, scanf and all that) with C++ based I/O
Functions in C++
Function overloading
Name mangling / decorating
Mixing C and C++ code modules (use of extern C)
Operators and operator overloading in C++