Programming with C++ 11 and 14
Duration: 3 days (9 hours each day with working lunch)

C++ is one of the few languages that continues to evolve in pleasantly surprising ways. The language supports lambda expressions, phenomenal features to improve performance and avoid some common errors, and includes a powerful multithreading library. This course will help you to quickly learn, apply, and exploit the capabilities in C++ 11 and 14.

The course has a good balance of interactive lectures and hands-on exercises. The attendees are expected to pair-up and work on the lab exercises. The instructor will assist the attendees as they work on the labs. The objective of the course is for the attendees to gain an in depth practical knowledge of the concepts so they can put them to immediate use on their real projects.

The course content will be customized to meet your teams' specific needs. Please review this detailed outline and suggest changes (additions, deletions, modifications) as you feel fit.

Topics

Functions and Expression Improvements
* Significance of advances
* Avoiding virtual trap
* override
* final
* noexcept
* conditional noexcept
* constexpr
* constexpr functions
* Exercise

Working with Typing
* Type checking
* Template Type Inference
* using auto
* auto and type inference
* auto in return type
* decltype
* power of decltype
* caution with type inference
* Exercise

Programming with Intention
* nullptr

* uniform initializer
* initializer list
* gotchas
* with auto
* with constructor
* Exercise

Functions and Types
* rvalue
* Forward references
* Reference qualifiers
* alias
* move vs. forward
* move semantics
* move constructor
* return value optimization
* caution
* removing defaults
* bringing in defaults
* enum improvements
* Exercise

Pointer Semantics
* Raw pointers
* Smart Pointers
* unique_ptr
* make functions
* shared_ptr
* weak_ptr
* using different pointer types
* Exercise

Other Advances:
* range based for-loop
* caution with looping
* type traits
* static_assert
* non-member begin/end
* creating custom begin/end
* local/unnamed types and templates
* callables
* Exercise

Programming with Lambdas
* Purpose and usage
* What are lambda expressions

* Higher-order functions
* Imperative vs. Functional Style
* defining lambdas
* Receiving Lambdas
* Lambdas vs. closures
* variable capture
* caution with capture
* Generic lambdas
* Exercise

Programming Concurrency
* Platform Neutral library
* level of concurrency
* creating threads
* managing shared resources
* thread functions
* passing data and gotchas
* Threading abstraction
* join and detach
* exceptions and threads
* using the RAII pattern
* Thread argument gotchas
* Dealing with mutability
* Rule for concurrency
* avoiding race conditions
* guards
* deadlock prevention
* multiple locks
* unique_lock
* using call_once and once_flag
* synchronization and conditional_variable
* Using timeouts
* Futures
* async
* packaged_task
* using promises
* Exercises

About the Instructor

Dr. Venkat Subramaniam is an award-winning author, founder of Agile Developer, Inc., creator of agilelearner.com, and an instructional professor at the University of Houston.

He has trained and mentored thousands of software developers in the US, Canada, Europe, and Asia, and is a regularly-invited speaker at several international conferences. Venkat helps his clients effectively apply and succeed with sustainable agile practices on their software projects.

Venkat is a (co)author of multiple technical books, including the 2007 Jolt Productivity award winning book Practices of an Agile Developer. You can find a list of his books at agiledeveloper.com. You may read more about Venkat and Agile Developer, Inc. at http://agiledeveloper.com.