

Test Driven Development in C++

Duration

2 days

Instructor

James Grenning

Class Limit

20 students

Price

On-site

Please contact SPC
for pricing (contact
information on page 2)

Public Training

\$1,695 (2 days)

*Discount available for
early registration

Materials Provided

- Student manual
containing the
course slides
- Student handouts
with class exercises

Test-driven development (TDD) is a software development technique that relies on the repetition of a very short development cycle: First the developer writes a failing automated test case that defines a desired improvement or new function, then produces code to pass that test and finally refactors the new code to acceptable standards. Kent Beck, who is credited with having developed or 'rediscovered' the technique, stated in 2003 that TDD encourages simple designs and inspires confidence.

Test-driven development requires developers to create automated unit tests that define code requirements before writing the code itself. The tests contain assertions that are either true or false. Passing the tests confirms correct behavior as developers evolve and refactor the code. Developers often use testing frameworks to create and automatically run sets of test cases..

This two-day workshop is a unique opportunity to learn from expert James Grenning the practice of Test Driven Development in C++. This course will get you and your team well on the way to applying TDD to your C++ applications.

Some comments from past participants:

"I was skeptical of TDD and had the concern that it would slow development down, but now I am sold 100% on TDD. Thanks again, the TDD training and your balanced look at agile / scrum was greatly appreciated!"

- Software Engineer

TRAINING

Test Driven Development in C++

Instructor

James Grenning, founder of Renaissance Software Consulting, trains, coaches and consults worldwide. With more than thirty years of software development experience, both technical and managerial, James brings a wealth of knowledge, skill, and creativity to software development teams and their management. He is currently writing a book on applying Test Driven Development to embedded software.

Articles by James have been published in IEEE software, IEEE Test and Measurement, and C++ Report. Areas of interest are software process improvement, Object Oriented Design, programming, embedded systems, project management, Extreme Programming, Test Driven Development, test automation and Agile software development. James knows his way around Scrum, with Scrum Master and Product Owner certifications.

Intended Audience

This seminar will be useful to software developers, technical team leaders, and managers that want to understand the technology they manage.

This two-day workshop, combining instructor-led discussion with practice sessions and multiple group exercises, is a hands-on course that teaches the practice of Test Driven Development in C++.

For more information on this or other SPC Springboard courses, please visit www.spcspringboard.com or e-mail SPC at info@spc.ca

Software Productivity Center Inc.
Suite 2000 1066 West Hastings Street
Vancouver, BC V6E 3X2

Vancouver: 604.662.8181 Toll Free: 1.877.548.1948

Fax: 604.689.0141

Outline

Test Driven Development

- Why Test Driven Development?
- What is Test Driven Development?
- TDD Microcycle
- CppUTest - unit test harness
- Exercise - Pairing with James
- Exercise - TDD from scratch

Testable Designs

- TDD and Collaborating Objects - testing the code in the middle
- Test Fakes, Stubs, Doubles, Mocks
- Exercise - Collaborating Objects
- Exercise - Self-verifying Mock Objects

Keeping Tests Clean

- Documentation value
- Ease of Maintenance
- Exercise - Clean up tests

Intro to Refactoring

- Test Smells, Code Smells
- Critical Skills
- Incremental Improvement
- Refactoring Plan
- Exercise - Long Method
- Exercise - Feature Envy
- Exercise - Large Class
- Exercise - Duplicate conditional logic

Architectural Vision and Incremental Delivery

- The Big Picture
- Just Enough Design
- Incremental Design
- Vertical Slices
- From Specific to General
- Design Principles

Wrap up Discussion



TRAINING