VECTOR

VectorCAST/C++ Training

Agenda VectorAcademy

Duration:	3 Days (delivered over 4 calendar days online)
Target Group:	Developers, Unit Test team, Integration Test team
Prerequisites:	Knowledge of C language, compiling/linking. C++ (if used by customer)
Goal:	Gain an understanding of using VectorCAST/C++ for Unit Testing and Code Coverage

1 | Introduction

- Terminology
- Unit test concepts
  - Basic unit test philosophy
  - Types of test cases

2 | Building Environments

- Prerequisites
  - What is required to use VectorCAST/C++
- Whitebox/Blackbox testing
- Environment building wizard
  - Steps to creating a VectorCAST/C++ testing environment
- Environment configuration
  - Configure items in VectorCAST Project

3 | Creating Test Cases

- Specifying input & expected values
- Data types
- Data entry techniques
- Compound tests
- Range and List Expressions

4 | Code Coverage

- Coverage types
- Coverage usage & options
- Coverage by Analysis
  - Augment coverage of hard to reach code
- MC/DC code coverage
  - Used in safety critical systems governed by DO-178, ISO 26262, etc.
  - Creating test cases based off MC/DC pairs

VECTOR

VectorCAST/C++ Training

Agenda VectorAcademy

5 | Test Reports

- Execution reports
  - Results of test case execution
- Coverage reports
- Management reports

6 | Regression Testing

- GUI usage only
- Basic configuration
- Analytics

7 | User Code

- Types of user code
  - Parameter user code
  - Test Case user code
  - Environment user code
- User code syntax

8 | Stubbing Fundamentals

- Stub by function
- Adding/Suppressing stubs
- Library stubs
- Controlling stub behavior

9 | Testing C++ Classes (C++ Customers only)

- C++ Harness architecture: class instance objects, member variables etc.
- Class instances and constructors

10 | Creating Tests from CSV Data

- Import CSV data to create tests
- Create import templates

11 | Generating Test Cases Using the Vector Test Data Editor (Windows Only)

- Generate automated test cases using a Classification Tree

VECTOR

VectorCAST/C++ Training

Agenda VectorAcademy

12 | System Testing with VectorCAST QA

- Creating a VectorCAST/QA environment
  - Configuring compiler, preprocessor and linker options
  - Instrumenting source code
- Automate using Python
  - Automate building instrumented source and test execution

13 | Requirements Gateway

- Importing requirements
- Exporting results to ALM/Requirements tools

14 | VectorCAST Workflows

- Establish VectorCAST workflow for unit and system test environments
- Best practices

15 | Probe Points

- Accessing local statics
- Testing hard to reach code

16 | Troubleshooting Build Process

- Options affecting build process
- Troubleshooting techniques in VectorCAST/C++

17 | Debugging Test Execution Problems

- Debugging techniques in VectorCAST
- Integrating source code debugger with VectorCAST/C++

18 | Advanced Troubleshooting Techniques

- Troubleshooting preprocess, compile and link errors