

Chapter 1: Getting Started

You will learn:

- Purpose of testing.
- Organization of testing effort.
- Requirements-based testing.
- Developing a test plan.
- Test plan approaches.
- Waterfall approach.
- Evolutionary approach.
- Unit testing.
- Integration testing.
- Regression testing.
- Building a library.

Chapter 2: Application Performance

You will learn:

- Why software fails.
- Improperly constrained input.
- Improperly constrained stored data.
- Improperly constrained computation.
- Improperly constrained output.
- Performance tuning.
- The tuning cycle.

Chapter 3: Code Coverage Analysis

You will learn:

- Purpose and features of code coverage.
- Structural testing and functional testing.
- Statement coverage.
- Decision coverage.
- Condition coverage.
- Multiple condition coverage.
- Condition/decision coverage.
- Modified condition/decision coverage.
- Path coverage.
- Coverage goal for release.

Chapter 4: Debugging

You will learn:

- Definition and purpose.
- Testing costs.
- Aspects of testing and debugging C code.
- Noticing and localizing a bug.
- Understanding a bug.
- Repairing a bug.
- Aspects of debugging C code.
- Types of bugs.
- C specific problems.
- Incremental building/linking.

Chapter 5: Testing Techniques

You will learn:

- Core dumps.
- Debugging techniques.
- Using the compiler's features.
- Warning options.
- Optimization flags.
- The RTFM technique.
- printf() debugging and testing.
- Guidelines.
- Assertions: defensive programming.
- Visual Studio Analyzer.
- Conditional termination.

TM

TM

Chapter 6: Testing Tools

You will learn:

- Editor.
- Version Management System.
- Debugger.
- Gbd.
- Memory Allocation Debugging Tools.
- System Call Tracers.
- Profilers.
- LINT.
- ANWB debugging.
- Code Walk Through.

TM

TM