

Polyspace for C/C++ Code Verification

OEM: MathWorks • Duration: 3 Days (24 hrs) • Code: MW-POLYSPACE-FOR-C-CPP-CODE-VERIFICATION

COURSE MODULES & TOPICS

Day 1 of 3

Polyspace Tool Comparison

- Code verification workflows
- Running a Bug Finder analysis
- Running a Code Prover verification
- Analyzing and fixing software defects and run-time errors

Polyspace Bug Finder Analysis

- Creating a Polyspace project
- Common run-time environment artifacts
- Handling processor-specific code
- Setting target hardware information
- Analyzing defects in Polyspace Bug Finder
- Analyzing MISRA C rule violations
- Measuring code metrics
- Sharing configuration settings

Analyzing Polyspace Code Prover Results

- Overview of abstract interpretation
- Analyzing the call tree
- Navigating through source code
- Interpreting Polyspace Code Prover results
- Viewing global variable usage in source code

Code Verification Checks

- Overview of C source code checks
- Polyspace Code Prover run-time error checks
- Relevant verification options
- Using workspaces

Day 2 of 3

Managing Polyspace Code Prover Verifications and Results

- Code Prover verification workflows
- Controlling program execution sequence
- Prioritizing orange run-time error checks

Adding Precision to Polyspace Code Prover Verifications

- Robustness verification and contextual verification
- Automatic stubbing
- Data range specification
- Manual stubbing

Integration Analysis with Polyspace Access

- Post-submit workflows with Polyspace Access
- Contextual verification
- Detecting integration defects and run-time errors

Application Analysis

- Application software concerns
- Analyzing concurrent data access issues
- Replacing assembly or hardware functions
- Handling third-party code in Polyspace
- Creating documentation

Day 3 of 3

Hands-On Instruction (Optional)

- Polyspace Bug Finder checks
- C++ code verification
- Tasking and shared data analysis
- Generated code verification
- Development process review
- Workflow integration
- Client/server software installation
- Polyspace configuration for project code
- Results interpretation

Appendix F: Integration Analysis with Polyspace Desktop

- Contextual verification
- Detecting integration defects and run-time errors