

Certified LabVIEW Developer (CLD) Exam Preparation Resource Guide

Preparation Guide Goal:

This guide offers resources to aid a candidate in preparing for a CLD exam. This document is by no means complete. Feedback and suggestions are solicited.

Exam Goal:

The CLD exam validates the knowledge and skills to design, develop and deploy a scalable, readable, and maintainable LabVIEW application using advanced software principles, architectures, techniques and the [LabVIEW Development Guidelines](#).

Exam Topics (Outline):

- Certified LabVIEW Associate Developer (CLAD) exam topics
- Software lifecycle models and design process
- VI design practices
- Project management
- Design patterns, event handling, and synchronization mechanisms
- Data structures and variants
- References and Property Nodes
- Multithreading, memory, and performance optimization
- Windows connectivity and communication
- Accessing external code
- VI Server
- Error handling
- Application deployment

Exam Prerequisite:

Valid CLAD certification.

Exam Details:

- Must be scheduled by contacting certification@ni.com
- Proctored by National Instruments staff
- 4-hour application development exam
- The following are provided or allowed during the exam:
 - A PC with LabVIEW. You may use the *LabVIEW Help* and LabVIEW documentation
 - Application development specification
 - Screenshot of front panel
 - Description of the controls and indicators
 - General requirements
 - Functional specifications of the application

Exam Grading:

- Applications are graded on three criteria:
 - Functionality
 - Programming Style
 - Documentation

Certified LabVIEW Developer (CLD) Exam Preparation Resource Guide

- Each of the criteria have approximately equal weight
- Passing grade is 75% or higher

Exam Grading Details:

The following points are taken into consideration for each of the grading criteria:

- Functionality:
 - Is the **Run** arrow broken?
 - Does the VI properly perform the requirements listed in the specifications?
 - Is the logic correct for Boolean inputs and outputs?
 - Does the VI respond to user inputs within the stated time limit (100ms)?
 - Does the VI / subVIs use 100% CPU time?
 - Is file I/O implemented correctly?
 - Does the application stop on error?

- Style:
 - Does the application follow [*LabVIEW Development Guidelines*](#)?
 - Is the VI:
 - Readable?
 - Constructed for scalability?
 - Easily maintainable?
 - Overly complex?
 - Is the VI constructed in a professional manner?
 - Does the VI use LabVIEW frameworks or design patterns?
 - Minimum requirement is to use a state machine
 - Is the VI hierarchical?
 - Repeated code should be in subVIs
 - Are type defined enumerated controls used to define states?
 - Does the VI use unnecessary temporary variables?
 - Are appropriate data types, ranges, and format/precision used for front panel controls?
 - Is data grouped in appropriate data structures: arrays or clusters?
 - Does the VI use deeply nested structures (2 or more)
 - Does the VI use sequence structures for purposes other than initialization or cleanup?
 - Does the VI use local and global variables?
 - Local variables can be used to update controls
 - Are global variables protected to avoid race conditions?
 - Are Property Nodes (value) used for updating indicators?
 - Are front panels and block diagrams well laid out?
 - Are block diagrams cramped into small spaces?
 - Are there unnecessary bends in wires?
 - Are objects / wires overlapping?
 - Are wires running under structures or structure borders?
 - Are the error terminals wired on VIs?

Certified LabVIEW Developer (CLD) Exam Preparation Resource Guide

- Are references closed appropriately?
- Is the VI optimized for memory and performance?
- Documentation
 - Is the VI documented through VI Properties?
 - Are the subVIs documented?
 - Are wires documented with appropriate labels?
 - Is the functionality documented?
 - Block diagram level
 - Main and nested structure level
 - Do front panel controls and indicators have descriptive names?
 - Do VIs have descriptive icons?
 - Are constants documented?
 - Do front panel controls have associated tip strips?
 - Does the top-level VI have a non-default icon?
 - Do all subVIs have consistent icon design?

Training / Tutorials:

- Paid training materials:
 - [National Instruments LabVIEW Intermediate I and II courses](#):
 - Instructor led
 - Self- paced by using the course manuals
 - [National Instruments CLD preparation course \(online\)](#)

The following table lists additional resources for specific topics:

Topic	Details of resources by topic
Functional global variables	● Global Variables
State machine design pattern	● Application Design Patterns: State Machines
Queued state machines	● Using Queues to Create a More Flexible State Machine
Event-driven programming	● Inside LabVIEW: Event-driven Programming
Event-based state machine	● Event-Driven Programming in LabVIEW
Advanced event handling techniques	● Advanced Event Handling with LabVIEW 7 Express
Queued message handler	● Queued Message Handler with Response Notification
Master/slave design pattern	● Application Design Patterns: Master/Slave
Producer/consumer design pattern	● Application Design Patterns: Producer/Consumer
VI Server-based design	● Dynamically Loading and Calling VIs
Synchronization,	● Optimizing Applications

**Certified LabVIEW Developer (CLD)
Exam Preparation Resource Guide**

multithreading, memory, and performance optimization	
Variants	<ul style="list-style-type: none">• Variant Data in LabVIEW: Mastering a Higher-Level Way to Work with Data
ActiveX	<ul style="list-style-type: none">• ActiveX
.NET	<ul style="list-style-type: none">• .NET
Networking	<ul style="list-style-type: none">• Networking
Accessing external code	<ul style="list-style-type: none">• Using External Code
Deploying applications	<ul style="list-style-type: none">• Distributing and Documenting VIs
Development tools and guidelines	<ul style="list-style-type: none">• Development Tools and Guidelines