

Customer Education Course Catalog



With a wide variety of professional resources for your personalized needs, including expert training and industry-recognized certifications, NI helps both beginners and experts get started on the next phase of their NI platform education.

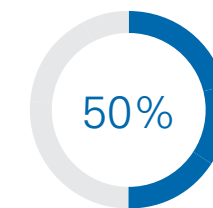


Learn Faster. Validate Skills. Accelerate Engineering.

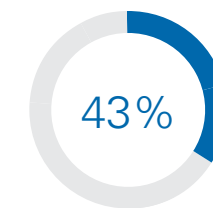
Whether you are new to NI products or have been using them for years, access to the right learning resources when you need them is critical to your success. NI provides a comprehensive customer education program to ensure success on your current application and at all stages of your career.

This program is the most effective way to increase productivity, reduce development time, and improve your ability to engineer robust, maintainable applications with NI products. It's also individually tailored to fit your schedule, work on your terms, and deliver core competency for beginners and experts alike.

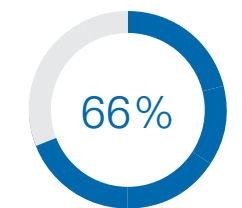
Customers who use training and certification services from NI report an average 66 percent faster learning times, 50 percent quicker development times, and 43 percent less time spent on maintenance after completing NI customer education.



Quicker Development Time



Less Time on Maintenance



Faster Learning Time

4
Overview

6
Learning Paths

16
Training Courses

22
Purchase Options

23
Learning Formats

24
Certifications

26
Course Details

“I estimate that the training courses have saved me more than 150 hours in self-paced learning time to get to an equivalent skill level.”

Thomas Sumrak, PAR Technologies, LLC

Follow your path. Find your success.

Increased productivity with NI tools starts with a clearly defined plan outlining your goals and application areas. You can then work with NI to identify the best strategies for acquiring and validating skills to take your productivity to the next level.

Learning paths are designed to help you navigate the learning resources available in your application area. Throughout the learning path, you can leverage assessments to identify the learning content you need to successfully develop applications.

Customize a learning path to your needs by choosing optional courses and topics, preferred learning format, and frequency of engagement with the materials.

LEARNING FORMATS

In-Product

Interactive tutorials in the software environment

Classroom

1- to 5-day classes at training facilities

On-Site

1- to 5-day classes at your office

Virtual

1 to 4 half-day live web-based classes

Online

Self-paced series of on-demand modules

Develop LabVIEW Applications

Go from developing basic applications in the LabVIEW graphical programming environment to architecting large applications in single- or multi-developer settings.

Design Smart Machines or Industrial Equipment

Learn how to rapidly design, prototype, and deploy embedded control and monitoring systems using LabVIEW and NI customizable off-the-shelf hardware.

Develop Automated Production Test Systems

Learn how to develop and manage test applications in the TestStand environment, distribute them to test stations, build test code for your hardware and devices under test, and log test results to a database.

Develop Hardware-in-the-Loop Test Systems

Develop a real-time test system and create a run-time editable user interface with VeriStand or the HIL and Real-Time Test Software Suite.

Measure Physical Systems With Sensors or Actuators

Acquire the skills to configure your DAQ hardware, perform accurate single-point and continuous measurements, and synchronize your data acquisition.

Semiconductor Test System Curriculum

Learn how to develop and debug configuration-based test programs, create custom measurements, and optimize advanced test programs for mixed-signal and RF devices using the NI Semiconductor Test System (STS).

Manage Data With DIAdem

Learn how to view and graphically investigate data, automate report generation, and extend the capabilities of DIAdem.

Acquire Images With Machine Vision

Learn to select and set up machine vision hardware and develop an application in LabVIEW that acquires, processes, analyzes, and inspects images.

Design Wireless Communications Systems

Learn how to rapidly design and prototype wireless communications systems with LabVIEW and software defined radio hardware.



Develop LabVIEW Applications

Reduce development time and costs through faster learning and increased productivity with LabVIEW. This learning path is for engineers developing applications using LabVIEW. It presents the courses, exams, and events to help you go from developing basic applications in the LabVIEW graphical programming environment to architecting large applications in single- or multiple-developer settings. Discover recommended techniques to successfully develop applications that can scale as your projects evolve. To learn more about training and certification for LabVIEW NXG, visit ni.com/training/labviewnxdg.

Getting Started

1

LabVIEW Core 1
+
LabVIEW Core 2
+
CLAD Exam

Explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format.

Design, implement, and distribute stand-alone applications using LabVIEW and apply single- and multiple-loop design patterns for application functionality.

Certified LabVIEW Associate Developers demonstrate a working knowledge of the LabVIEW environment, a basic understanding of coding and documentation best practices, and the ability to understand and interpret existing code.

Technician or basic developers using LabVIEW to develop, support, or troubleshoot a small to medium application can learn to apply basic design patterns and troubleshoot or debug LabVIEW code.

2

LabVIEW Core 3
+
CLD Exam

Learn structured best practices to design, implement, document, and test LabVIEW applications.

The second LabVIEW certification exam validates the ability to design and develop functional programs while minimizing development time and ensuring maintainability through proper documentation and style.

Software engineers who need to design and develop a medium to large application with more than one process will learn to evaluate file formats, create executables, handle errors during execution, and design to user requirements.

3

Managing Software Engineering
+
Advanced Architectures
+
CLA Exam

Effectively manage and deliver large LabVIEW applications in single- or multi-developer environments.

Discuss how to design and implement scalable, extensible software architectures for large LabVIEW applications.

The third and highest level of LabVIEW certification demonstrates mastery in architecting and project managing LabVIEW applications.

Software architects designing the application architecture or acting as technical lead or project manager for a large application can learn to leverage the appropriate tools and techniques to manage development; design, develop, and document a clean API; and analyze the architecture of a LabVIEW application for improvement.

Additional Options

LabVIEW Connectivity
LabVIEW Performance
Object-Oriented Design and Programming in LabVIEW
Actor-Oriented Design in LabVIEW

LabVIEW NXG Options

LabVIEW NXG Core 1
LabVIEW NXG Core 2
Transitioning to LabVIEW NXG

Proficiency Events

LabVIEW Developer Days
CLD Summit
CLA Summit

Badge Assessments

LabVIEW Programming
LabVIEW NXG Programming
LabVIEW Application Development
LabVIEW NXG Application Development

Common Learning Paths

Find the training courses, certification exams, and proficiency events to meet your application needs using the following customized learning paths. These paths were developed around the recommended skill levels and additional areas of interest someone with this application might have.

Use the descriptions to the right of the learning path to help you determine how far along the path to progress. Keep in mind many of our courses are available in person or on demand. For your consideration, optional courses, exams, events, and assessments have been included at the bottom of each path and can be used anytime to help you customize your learning to your needs.

Design Smart Machines or Industrial Equipment

This learning path is for users developing embedded control and monitoring systems to design smart machines or industrial equipment. It presents courses, exams, and events that teach you how to design, prototype, and deploy reliable and deterministic embedded systems using LabVIEW. Consider this path if you are using the Embedded Control and Monitoring Software Suite, the LabVIEW Real-Time Module, or the LabVIEW FPGA module with CompactRIO, Single-Board RIO, PXI, or R Series multifunction reconfigurable I/O devices.

Getting Started

1

LabVIEW Core 1
+
LabVIEW Core 2
+
CLAD Exam

Explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format.

Design, implement, and distribute stand-alone applications using LabVIEW and apply single- and multiple-loop design patterns for application functionality.

Certified LabVIEW Associate Developers demonstrate a working knowledge of the LabVIEW environment, a basic understanding of coding and documentation best practices, and the ability to understand and interpret existing code.

For those designing, prototyping, and deploying embedded control and monitoring applications, LabVIEW Core courses introduce the LabVIEW environment, communication mechanisms, and development techniques.

2

Embedded Control and Monitoring
+
LabVIEW for CompactRIO Developer's Guide
+
CLEED Exam*

Learn how to efficiently design, prototype, and deploy a reliable embedded control and monitoring application.

Get an overview of recommended architectures and development practices when programming CompactRIO controllers.

Certification demonstrates proficiency and expertise in analyzing requirements for and designing, developing, debugging, and deploying reliable mission-critical embedded control and monitoring applications.

Translate system requirements into a scalable software architecture, choose appropriate methods for interprocess and network-based communication, design for reliability, and efficiently deploy and replicate your embedded system.

*CLEED exam requires a valid CLD or CLA certification to attempt.

Additional Options

LabVIEW Core 3
CLD Exam
LabVIEW Machine Vision

"I'm more confident in my abilities knowing I'm part of a select group with proven skills. Now that I am certified, I'm more respected at work, I'm more respected by customers, and I'm more respected by attendees of my local user group. I maintain my certification for continued invitations to the CLA Summit, customer reassurance, my public profile, and to be the best I can at what I do."

Richard Thomas, Thoric Solutions Ltd, CLA, CTD, and CLED

Develop Automated Production Test Systems

This learning path is for engineers building manufacturing test or production test applications with the Automated Test Software Suite or TestStand. It presents the courses, exams, and events that teach you to develop practical test applications in the TestStand environment and distribute them to test stations, build test code for your hardware and devices under test, and log test results to a database.

Getting Started

1

LabVIEW Core 1
or
LabWindows™/CVI
Core 1
+
Hardware Training

Explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format.

Develop application solutions using the built-in features of the LabWindows/CVI ANSI C integrated development environment.

Hardware training: Learn to install, configure, and program your NI hardware with a wide variety of available training courses.

Test developers building code modules for individual components of a DUT or defining hardware connections for the test station should familiarize themselves with the programming environment and hardware functions through the relevant training courses.

2

Developing Test Programs Using TestStand
+
CTD Exam

Use TestStand to develop, analyze, debug, and deploy practical test applications that match your test needs.

Certified TestStand Developers demonstrate the ability to develop, debug, and deploy functional applications in TestStand software while minimizing development time and ensuring maintainability.

Technicians and test system designers using TestStand as their test executive develop, execute, and debug an automated test system as well as manage, package, and deploy test software.

3

Architecting Test Systems Using TestStand
+
CTA Exam

Design and architect test system frameworks, learn advanced features, and customize the built-in functionality of TestStand.

Certified TestStand Architects display expertise in architecting test systems based on TestStand from high-level specifications.

Test software architects define and create the software framework for the test organization, develop or select data and user management systems, and develop customized and robust user interfaces.

Additional Options

LabVIEW Core 2
CLAD Exam
LabVIEW Instrument Control
Data Acquisition Using NI-DAQmx and LabVIEW

LabWindows/CVI Core 2
Modular Instrument Courses

Badge Assessments

TestStand Navigation and Diagnostics
TestStand Sequence Development
Test Principles
Test Instrumentation

Develop Hardware-in-the-Loop Test Systems

This learning path is for engineers developing real-time test and hardware-in-the-loop applications with VeriStand or the HIL and Real-Time Test Software Suite. It presents the courses, exams, and events that teach you how to develop a real-time test system and create a run-time editable user interface using VeriStand.

Getting Started

1 HIL Fundamentals Using NI VeriStand

Use the out-of-the-box features of VeriStand to develop real-time test applications such as hardware-in-the-loop test systems.

Whether you are creating real-time test systems such as HIL simulators and dynamometers or are tasked with executing and modifying existing VeriStand test systems, you can learn the fundamentals for test development and operation.

2 LabVIEW Core 1 + Embedded Control and Monitoring

Explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format.

Learn to add more functionality to VeriStand using the power of LabVIEW. Whether you need to create custom real-time logic, develop hardware algorithms on an FPGA, or redesign your user interface, LabVIEW training courses will provide you with useful skills to add functionality to your real-time test system.

3 DIAdem Basics or Developing Test Programs + CTD Exam

Learn to use the most important analysis, reporting, and data management features of the DIAdem environment.

Use TestStand to develop, analyze, debug, and deploy practical test applications that match your test needs.

Certified TestStand Developers demonstrate the ability to develop, debug, and deploy functional applications in TestStand software while minimizing development time and ensuring maintainability.

Improve the efficiency of your real-time test application by incorporating advanced automation, post-processing, and report generation. Learn to create automation scripts, custom test reports, and data analysis algorithms using TestStand and DIAdem.

Additional Options

LabVIEW Core 2
CLAD Exam
LabVIEW Core 3

CLD Exam

Data Acquisition Using NI-DAQmx and LabVIEW
LabVIEW for CompactRIO Developer's Guide

Badge Assessments

LabVIEW Programming
DIAdem Data Exploration

Measure Physical Systems and Acquire Data

This learning path is for engineers using NI DAQ devices and LabVIEW to develop customized and automated measurements to meet application requirements. It presents the courses, exams, and events that teach you how to configure your hardware, perform accurate single-point and continuous measurements, and synchronize your data acquisition.

Getting Started

1 LabVIEW Core 1 + Data Acquisition Using NI-DAQmx and LabVIEW + LabVIEW Core 2 + CLAD Exam

Explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format.

Explore the fundamentals of data acquisition using sensors, NI data acquisition hardware, and LabVIEW.

Design, implement, and distribute stand-alone applications using LabVIEW and apply single- and multi-loop design patterns for application functionality.

Certified LabVIEW Associate Developers demonstrate a working knowledge of the LabVIEW environment, a basic understanding of coding and documentation best practices, and the ability to understand and interpret existing code.

After an introduction to the LabVIEW environment, communication mechanisms, and development techniques, developers learn to connect signals; acquire, display, and log measurements; generate waveforms; and select resolution and sample rates.

2 LabVIEW Core 3 + CLD Exam

Learn structured best practices to design, implement, document, and test LabVIEW applications.

The second certification exam validates the ability to design and develop functional programs while minimizing development time and ensuring maintainability through proper documentation and style.

For those who need to integrate data acquisition as a component in a large application or add advanced functionality such as synchronization or handling large quantities of data, advanced LabVIEW training is recommended.

Additional Options

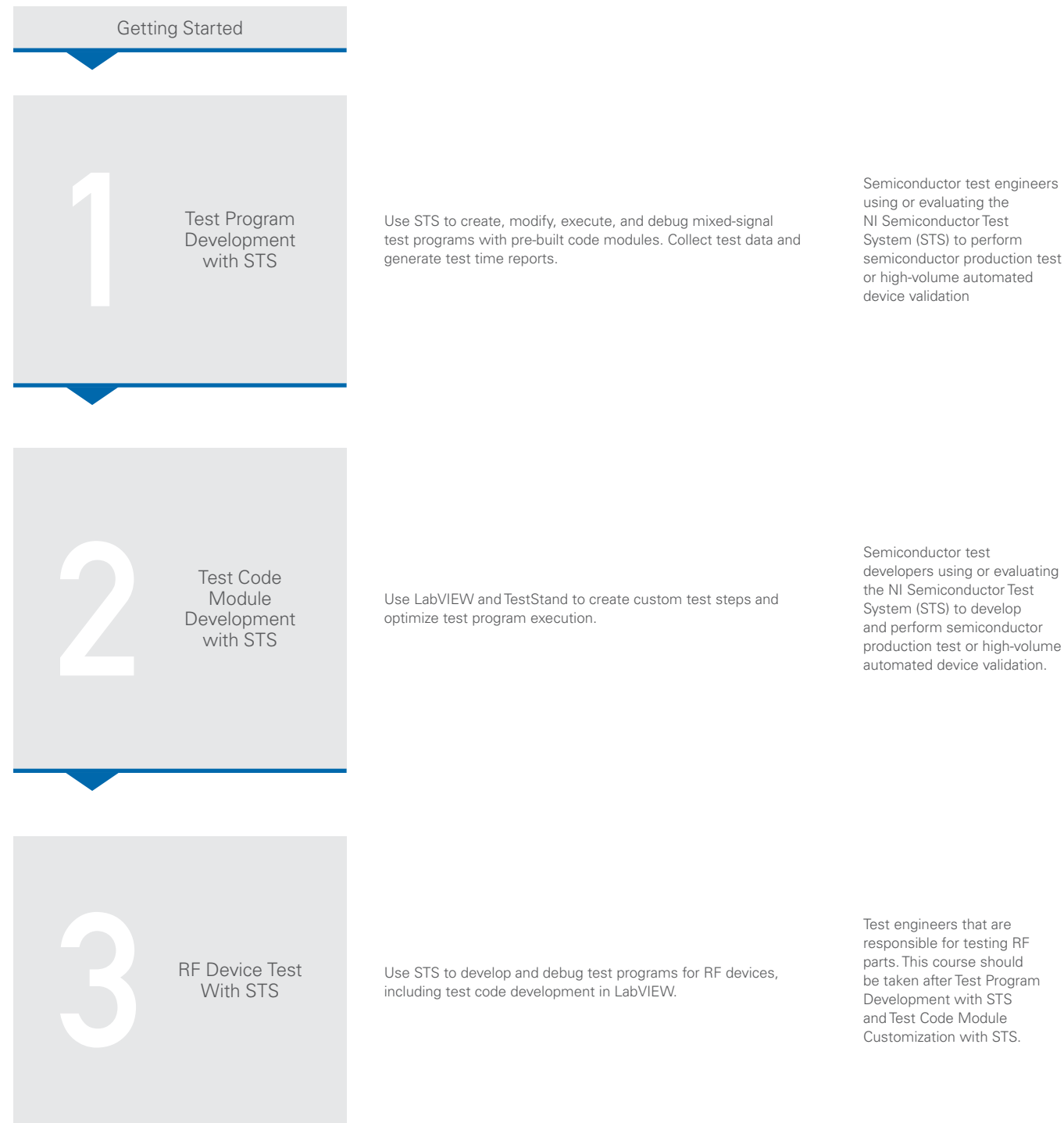
Data Acquisition Using LabVIEW NXG
LabWindows/CVI Core 1
LabWindows/CVI Core 2
Sound and Vibration Fundamentals
DIAdem Basics
LabVIEW Instrument Control

Badge Assessments

Taking Physical Measurements
LabVIEW Programming
LabVIEW NXG Programming

Semiconductor Test System Curriculum

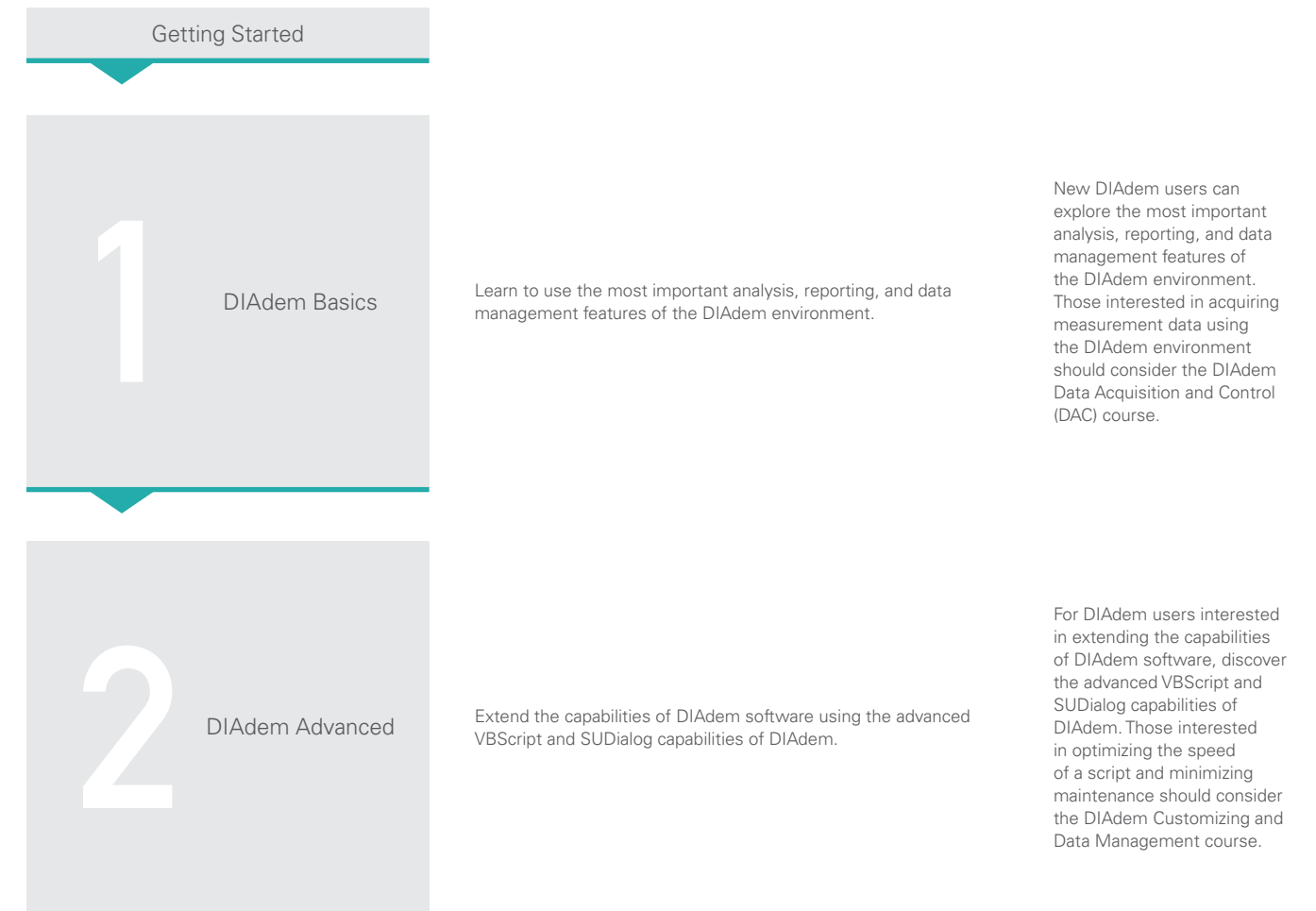
Semiconductor production test engineers are often challenged to test more complex parts in a fraction of the time and budget. The STS Test Engineer Curriculum is a learning path consisting of three courses designed to quickly teach semiconductor production test engineers how to develop and debug configuration-based test programs, create custom measurements, and optimize advanced test programs for mixed-signal and RF devices using the NI Semiconductor Test System (STS).



Additional Options
 LabVIEW Core 2
 Developing Test Programs Using TestStand

Manage Data With DIAdem

This learning path is for engineers using DIAdem to quickly locate, inspect, analyze, and report on measurement data. It presents the courses, exams, and events that teach you how to import, view, and graphically investigate data; automate report generation; and extend the capabilities of DIAdem.



Additional Options
 DIAdem Data Acquisition and Control
 LabVIEW Core 1

Badge Assessments
 DIAdem Data Exploration

Acquire Images With Machine Vision

This learning path is for engineers developing vision applications with NI machine vision hardware and software. It presents the courses, exams, and events to help you select and set up the machine vision hardware and develop an application in LabVIEW that acquires, processes, analyzes, and inspects images for applications such as assembly verification, object tracking, and vision-guided robotics.

Getting Started

1

LabVIEW Core 1

+

LabVIEW Machine Vision

+

LabVIEW Core 2

+

CLAD Exam

Explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format.

Learn the fundamentals of machine vision tasks and inspections. Select and use the hardware components of a machine vision system.

Design, implement, and distribute stand-alone applications using LabVIEW and apply single- and multi-loop design patterns for application functionality.

Certified LabVIEW Associate Developers demonstrate a working knowledge of the LabVIEW environment, a basic understanding of coding and documentation best practices, and the ability to understand and interpret existing code.

After an introduction to the LabVIEW environment, communication mechanisms, and development techniques, basic vision system developers learn hardware selection, setup, and use and the acquisition, display, and processing of images.

2

LabVIEW Core 3

+

CLD Exam

Learn structured best practices to design, implement, document, and test LabVIEW applications.

The second certification exam validates the ability to design and develop functional programs while minimizing development time and ensuring maintainability through proper documentation and style.

As your applications become more complex, advanced LabVIEW training is recommended to learn best practices for the LabVIEW graphical programming environment.

Badge Assessments
 LabVIEW Programming
 LabVIEW NXG Programming

“On-site training is an excellent solution for customers that need to train multiple people. It offers a level of flexibility in the delivery of that training that personalizes it for your unique needs.”

Trey McWhorter, NI Account Representative to Hach Company

Design Wireless Communications Systems

This learning path is for engineers designing and prototyping wireless communications systems. It presents the courses, exams, and events that teach you fundamentals to rapidly bring your wireless idea to life with the LabVIEW Communications System Design Suite and NI software defined radio hardware.

Getting Started

1

LabVIEW Communications Guided Help

Learn about interactive materials in the LabVIEW Communications development environment that introduce software concepts.

LabVIEW Communications programmers can familiarize themselves with the software environment and learn the basics of navigating the design flow and understanding specific design languages.

After an introduction to the LabVIEW environment, communication mechanisms, and development techniques, basic vision system developers learn hardware selection, setup, and use and the acquisition, display, and processing of images.

2

LabVIEW NXG Core 1

+

LabVIEW NXG Core 2

+

LabVIEW FPGA

Design, implement, and distribute stand-alone applications using LabVIEW NXG and apply single- and multi-loop design patterns for application functionality.

Explore the LabVIEW NXG environment, dataflow programming, and common LabVIEW NXG development techniques in a hands-on format

Design, debug, and implement efficient, optimized applications using the LabVIEW FPGA Module and reconfigurable I/O (RIO) hardware.

After an introduction to the LabVIEW NXG environment, communication mechanisms, and development techniques, engineers learn FPGA techniques and best practices

3

LabVIEW Communications Application Frameworks

Explore ready-to-run, easily modifiable reference designs included with the LabVIEW Communications software. Application frameworks included are 802.11 and LTE.

A substantial starting point for researchers working to improve communications standards is modifying source code add-ons to more rapidly innovate on new prototypes with LabVIEW Communications.

As your applications become more complex, advanced LabVIEW training is recommended to learn best practices for the LabVIEW graphical programming environment.

| TRAINING COURSES | LEARNING FORMATS | | | | LEARNING PATHS | | | | | | | | |
|---|------------------|---------|---------|--------|------------------------------|---|---|---|--|--------------------------------------|-------------------------|------------------------------------|--|
| | Classroom | On-Site | Virtual | Online | Develop LabVIEW Applications | Design Smart Machines or Industrial Equipment | Develop Automated Production Test Systems | Develop Hardware-in-the-Loop Test Systems | Measure Physical Systems With Sensors or Actuators | Semiconductor Test System Curriculum | Manage Data With DIAdem | Acquire Images With Machine Vision | Design Wireless Communications Systems |
| Actor-Oriented Design in LabVIEW | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● | ● |
| Advanced Architectures in LabVIEW | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Architecting Test Systems Using TestStand | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Data Acquisition Using LabVIEW NXG | ● | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● |
| Data Acquisition Using NI-DAQmx and LabVIEW | ● | ● | ● | ● | ● | ● | ○ | ○ | ● | ● | ● | ● | ● |
| Developing Test Programs Using TestStand | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| DIAdem Basics | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| DIAdem Advanced | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| DIAdem Data Acquisition and Control | by request | ● | ● | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ● |
| Embedded Control and Monitoring Using LabVIEW | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| FlexRIO | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| High-Throughput LabVIEW FPGA | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| HIL Fundamentals Using NI VeriStand | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Communications 802.11 Application Framework | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Communications LTE Application Framework | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Connectivity | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Core 1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Core 2 | ● | ● | ● | ● | ● | ● | ○ | ○ | ● | ● | ● | ● | ● |
| LabVIEW Core 3 | ● | ● | ● | ● | ● | ○ | ● | ○ | ● | ● | ● | ● | ● |
| LabVIEW for CompactRIO Developer's Guide | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● |
| LabVIEW FPGA | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Instrument Control | ● | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● |

| TRAINING COURSES | LEARNING FORMATS | | | | LEARNING PATHS | | | | | | | | |
|---|------------------|---------|---------|--------|------------------------------|---|---|---|--|--------------------------------------|-------------------------|------------------------------------|--|
| | Classroom | On-Site | Virtual | Online | Develop LabVIEW Applications | Design Smart Machines or Industrial Equipment | Develop Automated Production Test Systems | Develop Hardware-in-the-Loop Test Systems | Measure Physical Systems With Sensors or Actuators | Semiconductor Test System Curriculum | Manage Data With DIAdem | Acquire Images With Machine Vision | Design Wireless Communications Systems |
| LabVIEW Machine Vision | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW NXG Core 1 | ● | ● | ● | ● | ○ | ● | ● | ● | ○ | ● | ● | ● | ● |
| LabVIEW NXG Core 2 | ● | ● | ● | ● | ○ | ● | ● | ● | ○ | ● | ● | ● | ● |
| LabVIEW Performance | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Real-Time 1 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabVIEW Real-Time 2 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| LabWindows/CVI Core 1 | ● | ● | ● | ● | ● | ● | ● | ○ | ○ | ○ | ● | ● | ● |
| LabWindows/CVI Core 2 | ● | ● | ● | ● | ● | ● | ● | ○ | ○ | ○ | ● | ● | ● |
| Managing Software Engineering in LabVIEW | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Modular Instruments 1: Digital Multimeters | by request | ● | ● | ● | ● | ● | ● | ○ | ○ | ● | ● | ● | ● |
| Modular Instruments 2: Switches | by request | ● | ● | ● | ● | ● | ● | ○ | ○ | ● | ● | ● | ● |
| Modular Instruments 3: High-Speed Digital I/O | by request | ● | ● | ● | ● | ● | ● | ○ | ○ | ● | ● | ● | ● |
| Multisim Basics | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Object-Oriented Design and Programming in LabVIEW | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● | ● |
| RF Device Test With STS | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Sound and Vibration Fundamentals | by request | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● |
| Test Code Module Development With STS | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Test Program Development With STS | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Transitioning to LabVIEW NXG | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● | ● |
| Ultiboard Basics: PCB Layout | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Using LabVIEW and TestStand in Regulated Industries | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ● |
| Using NI InsightCM™ for Condition Monitoring | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |


Accelerate learning and promote your accomplishments with NI badges.

Find learning resources and gain skills related to your application. Track your knowledge growth with milestone badge assessments and professional certifications. Then share your successes with others on social media and job sites.

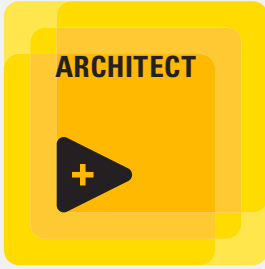
What Is a Badge?

A badge is a digital credential that NI issues to you in recognition of your accomplishments, such as passing an assessment, exam, or audit. These digital credentials contain verified, personalized information, including details on the exam you passed to earn your certification or your skills tested in an application area.


NI Badges



Learning Badges
Discover engineering fundamentals and NI product best practices through on-demand learning and then test your understanding with self-assessments and share your success.



Professional Certification
Demonstrate that you have the skills needed to create high-quality applications with NI software. Share your skills on social media and job sites and advance your career.



Community Advocacy
Gain recognition for technical leadership, participation in, and contributions to the NI user community and NI's expansive ecosystem of tools and expert developers.

Promote Your Accomplishments

You can hyperlink a badge image file to the digital credential containing your unique, verified data. This makes badges more secure than paper-based certificates and eliminates the possibility of anyone claiming your credential. Adding a hyperlinked badge image to your email signature and sharing your badge on social media and job sites are great ways to make sure your professional network is aware of your accomplishments.

Benefits of Badges

- You receive feedback on your learning and uncover next steps in your skill progression
- Employers, peers, and customers can instantly verify your skills
- You can easily share your achievements on a variety of social media platforms

Have questions? Email BadgeManager@ni.com.

Challenge yourself and test your understanding with NI learning badges.

Complement NI courses and certifications with learning badge self-assessments to track your and knowledge growth and find additional content related to your application. Test your understanding of engineering fundamentals and best practices using NI products with these free online assessments.

Use Learning Badges to:

- Prepare for NI certification exams
- Identify knowledge gaps before starting a new project
- Optimize your self-paced learning and target topics specific to your needs

How Do Learning Badges Work?

1. Select a badge related to your projects and review the available learning resources
2. Take the free, open-book, untimed assessment
3. Earn all the badges in a badge learning path and receive the high-level badge for that path
4. Follow instructions from Acclaim to share your badges with others on social media and job sites

Choose a badge learning path that best fits your application needs.



Automate simple tasks and take basic measurements on a one-off basis. Build a simple looping or sequencer VI from scratch. (*LabVIEW* and *LabVIEW NXG*)



Take ad-hoc engineering measurements using LabVIEW and NI data acquisition hardware. Build a custom measurement solution to acquire and visualize real-world signals.



Build a foundation of test and measurement principles to help you test multiple products with a test platform.

Visit ni.com/badges to take a free learning badge assessment.



Personalize with multiple learning formats.

NI offers courses in several languages and formats, including classroom training at facilities worldwide or on-site at your facility, online courses, and virtual training to better serve your individual needs. Whichever course format you choose, NI training courses can help you achieve immediate productivity gains and long-term success. Visit ni.com/training/options.

I In-Product

Start learning as soon as you open your project with guided help built into the development environment. Learn how to navigate a new software environment with a series of steps, theory, and interactive elements.

O Online

Learning with on-demand modules is a cost-effective alternative that is accessible 24 hours a day. Each online training course is an interactive, multimedia experience that covers key concepts over a series of modules. Online training can also serve as a quick reference on technical topics or preparation for an upcoming certification exam.

V Virtual

With one to four half-day live classes online, you can reduce your training and development costs by eliminating travel and time away from work. Virtual learning combines live instruction with remote access to software and hardware over the Internet for a productive, lasting experience.

C Classroom

With one- to five-day classes at training facilities around the world, you can leverage one of the most effective forms of learning. Classroom training requires an investment of time and effort but provides significant rewards, including networking with instructors and peers, an escape from office distractions, and hands-on access to hardware for exercises.

S On-Site

With one- to five-day classes for several employees at your office or location of choice, you can eliminate travel and hotel expenses. On-site training also gives you the opportunity to offer courses tailored to your company's specific needs. Spend time on topics of interest and standardize skills sets across your team.

Make a sound investment with purchasing options.

NI training teaches you recommended techniques to reduce development time and improve application performance and scalability. With courses taught by experienced, certified instructors, you can rest assured that your upfront investments in training result in significant labor cost savings for projects and nearly 3X return on investment.

And, whether you are ready to enroll now or preparing for the future, NI offers a variety of cost-effective purchasing options to best meet your needs. Visit ni.com/training/buy.

50%

Quicker Development

43%

Less Maintenance

66%

Faster Learning



Save Money With Membership

Interested in multiple training courses or exams? Membership is a cost-effective way to take all the training you want. Choose from six-month, one-year, and two-year options.



Buy Training Credits Now, Schedule Later

You can purchase training credits now and redeem them later for any training or certification offering. Valid for one year, credits can be applied to anyone, not just the purchaser.



Enroll in a Course Now

Already know the course or certification exam you are looking for and when you would like to take it? Start browsing our training catalog now.



Buy Software, Get Training

An active software service contract includes 24/7 access to online training. The Software Suites include a one-year training membership for unlimited access to live training.

Already a Member of the Standard Service Program (SSP)?

With an active LabVIEW SSP membership, you are entitled to online training. Visit ni.com/ssp to see the courses you are eligible for and begin viewing training videos now.

Validate your skills and advance your career with certifications.

Certification demonstrates you have the skills needed to create high-quality applications with NI software and gives customers, peers, and employers confidence in your abilities. With nearly 20,000 certified professionals worldwide, NI certifications are widely recognized in industry. NI offers certifications that are standardized globally, allowing for secure and uniform testing across teams, organizations, or countries. You can use certification to assess and validate an individual's skills for the purpose of project staffing or career advancement.

CLAD | Certified LabVIEW Associate Developer

The CLAD certification indicates a broad working knowledge of the LabVIEW environment, a basic understanding of coding and documentation best practices, and the ability to read and interpret existing code.

CLD | Certified LabVIEW Developer

The CLD certification indicates the ability to design and develop functional programs while minimizing development time and ensuring maintainability through proper documentation and style.

CLA | Certified LabVIEW Architect

The CLA exam tests the user's ability to build a sensible VI hierarchy and project plan for delivering an application that meets project requirements. Certified Architects can design the application architecture and manage the development of the individual components by other engineers.

CLED | Certified LabVIEW Embedded Systems Developer

A CLED demonstrates expertise in designing, developing, debugging, and deploying applications based on CompactRIO, Single-Board RIO, and/or R Series hardware. A CLED efficiently uses the LabVIEW Real-Time and LabVIEW FPGA modules with NI-recommended best practices to design modular, scalable, and maintainable embedded systems to requirements.

CTD | Certified TestStand Developer

The first step in the two-part TestStand certification process, the CTD measures the ability to develop, debug, and deploy functional applications in TestStand software while minimizing development time and ensuring maintainability.

CTA | Certified TestStand Architect

The final step in the two-part TestStand certification process, the CTA shows a user's expertise in architecting test systems in TestStand based on high-level specifications. Certified Architects can design the application architecture and manage the development of individual components by other engineers.

CCVID | Certified LabWindows/CVI Developer

A CCVID demonstrates a solid understanding of the LabWindows/CVI environment, core features, and its functionalities.

Certified Professional Instructor

The CPI certification distinguishes individuals who are authorized to teach NI courses. CPIs are the only non-NI instructors authorized to teach NI courses. Through teaching courses, CPIs can identify business opportunities and position themselves as experts in their field.



| CERTIFICATION EXAM FORMATS | Multiple Choice | Practical | Prerequisites | Exam Duration | Recertification Interval |
|--|-----------------|-----------|-------------------------|---------------|--------------------------|
| Certified LabVIEW Associate Developer | ● | ● | NONE | 1 HOUR | 2 YEARS |
| Certified LabVIEW Developer | ● | ● | NONE | 4 HOURS | 3 YEARS |
| Certified LabVIEW Architect | ● | ● | CLD | 4 HOURS | 4 YEARS |
| Certified LabVIEW Embedded Systems Developer | ● | ● | CLD OR CLA | 1 HR, 5 HR | 5 YEARS |
| Certified TestStand Developer | ● | ● | NONE | 1 HOUR | 3 YEARS |
| Certified TestStand Architect | ● | ● | CTD | 1 HOUR | 4 YEARS |
| Certified LabWindows/CVI Developer | ● | ● | NONE | 1 HOUR | 3 YEARS |
| Certified Professional Instructor | ● | ● | CLD/CTD/CCVID OR HIGHER | 8 HOURS | SUBJECT TO REVIEW |

Prepare for an Exam

Preparing for your exam is easy with live and on-demand resources such as preparation guides, webcasts, sample exams, and solutions. See a recommended preparation plan customized to your experience and the time you have until your scheduled exam.

Visit ni.com/certification-prep to learn more.

Register for an Exam

NI's multiple choice certifications are offered at Pearson VUE testing centers worldwide. Visit pearsonvue.com/ni to schedule your exam. For practical certifications, contact your local NI office or email certification@ni.com to register.

Visit ni.com/certificationschedule for more details.

Explore LabVIEW NXG Certification options at ni.com/training/labviewnxdg.

"You can be confident in what you are getting if that certification level is there. Those individuals have the knowledge and experience to hit the ground running, and we had an aggressive timetable for completing the project."

— David Hakey, GE Energy

Not Quite Ready for an Exam?

If you are not quite ready for a certification exam, discover what you need to learn through the NI Badge Program. Assess your skill level with NI Badge assessments and view recommended learning material as you progress through a badge path.

Visit ni.com/badges to learn more.

Actor-Oriented Design in LabVIEW

Learn to use the Actor Framework to design and implement scalable, extensible software architectures for LabVIEW applications requiring a high level of concurrency. Participate in discussions and work independently and collaboratively to design and build individual actors and to architect an application built with those actors. Discover how to work in the framework, use inheritance and composition to build highly parallel systems with rich behavior, and test and deploy completed systems.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Build actors using the Actor Framework
- Design and build complex, multi-actor systems to solve problems requiring a high level of concurrency
- Examine common design patterns in actor systems to promote reuse and extensibility

Prerequisites

- LabVIEW Core 3
- Object-Oriented Design and Programming in LabVIEW

NI Products Used

- LabVIEW Professional Development System
- LabVIEW Desktop Execution Trace Toolkit

Coursework Topics

Actor Framework: Fundamentals

Get an introduction to the Actor Framework and explore the development challenges it helps solve.

Creating Actors

Discover the process for creating new actors, from user interfaces to nested actors.

Introduction to the Course Project

Discuss several possible design approaches for solving a given challenge with both traditional LabVIEW and an actor-oriented design solution.

Building Actors Through Inheritance

Discover how you can use inheritance to implement complex behavior in a single actor.

Hardware Abstraction Layers

Learn the value hardware abstraction layers (HALs) offer your application development and ways to implement HALs using the Actor Framework.

Building Actors Through Composition

Discover how you can design an actor that uses other actors to implement complex behavior.

Zero Coupling and Abstract Messages

Send messages from a nested actor to its caller without knowing anything about the caller.

Designing User Interfaces

Explore two options for creating user interfaces in the Actor Framework: (1) through inheritance and (2) based on Model-View-Controller. Learn to select and implement the best option for the problem at hand.

Distributed Systems

Design actors for distributed systems. Learn how to connect actors to FPGA code on multiple targets and how to manage UIs for multiple subsystems.

State Pattern Actors

Examine the state pattern and how you can implement it using the Actor Framework.

Advanced Message Handlers

Learn how you can change the way actors handle messages and when to consider techniques such as using message filters, redirecting messages, and generating custom trace events.

Actors in Synchronous Systems

Explore techniques for integrating actors into a synchronous system. Call an actor from a synchronous environment and use reply messages.

Automated Test

Discover how to design a test harness to perform automated test on an actor.

Suggested Next Courses

- Advanced Architectures in LabVIEW
- Managing Software Engineering in LabVIEW

Suggested Badge Assessments

Actor Framework

- Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



Advanced Architectures in LabVIEW

The Advanced Architectures in LabVIEW course discusses designing and implementing scalable, extensible software architectures for large LabVIEW applications. During the course, participate in discussions and work independently and collaboratively to learn how to architect an application and design components to support the architecture. Also learn about several advanced design patterns, reference architectures, interprocess communication methods, and more.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Refine a requirements document and design a scalable, readable, maintainable, and extensible software architecture for a large LabVIEW-based application
- Collaborate with a team to create an architecture
- Understand advanced design patterns and how to use them to implement the components or subsystems of an architecture
- Evaluate various methods of communicating between multiple processes
- Understand the design trade-offs when selecting an advanced design pattern and interprocess communication method

Prerequisites

- LabVIEW Core 3 or equivalent experience

NI Products Used

- LabVIEW Professional Development System Version 2011 or later

Coursework Topics

Architecting an Application

Learn how to design and document scalable, readable, and maintainable software architectures. Topics include the characteristics of a scalable, readable, and maintainable architecture; software architecture documentation; and characteristics of a scalable design pattern.

Designing an API

Explore how to design a consistent, organized, and usable API that may be reused in your software architectures and distributed to several developers. Topics include API design techniques, polymorphic VIs for an API, project libraries for API design, and data passing in an API.

Multiple Processes and Interprocess Communication

Learn about foundational APIs and design patterns and how they apply to several essential advanced design patterns. Also examine several advanced methods for communicating between multiple processes and their trade-offs. Exercises and tools are general enough that you can use them in your own applications.

Advanced User Interface Techniques

Discover how to create an architecture that provides a modular, scalable, and extensible user interface. Topics include subpanels and XControls.

Introduction to Object-Oriented Programming in LabVIEW

See how using the encapsulation and inheritance capabilities of LabVIEW object-oriented programming can create scalable and extensible software architectures. Topics include using encapsulation to restrict access to class data and using inheritance and dynamic dispatch to implement scalable polymorphism at run time.

Plug-In Architectures

Examine how to create a plug-in architecture that you can use to add features without changing your main code. Topics include plug-in architectures using the VI Server, plug-in architectures using LabVIEW object-oriented programming, and subVI creation from an existing VI.

Tips, Tricks, and Other Techniques

Explore various tips, tricks, and techniques to help create your architecture. Topics include variant attributes, callback VIs with user events, VI scripting, and drop-in VIs.

Error Handling

Learn how to design a comprehensive error-handling strategy that can classify errors and take specific and general actions for errors. Topics include error-handling techniques that respond to specific errors, central error-handling techniques that respond to classes of errors, and an example LabVIEW error-handling reference library.

Additional Resources

Discover where to find additional code and ideas including through NI community groups and other forums.

Suggested Next Courses

- Managing Software Engineering in LabVIEW
- Object-Oriented Design and Programming in LabVIEW
- Actor-Oriented Design
- LabVIEW Performance

Validate Your Expertise With an NI Certification

Certified LabVIEW Architect (CLA)

Prerequisite: CLD
 Format: Application development
 Duration: 4 hours
 Location: NI branch office, training center, online, or on-site at your location

Recertification Process

Recertification Interval: 4 years

- Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Architecting Test Systems Using TestStand

Building on the Developing Test Programs Using TestStand course, the Architecting Test Systems Using TestStand course teaches you how to design and architect test system frameworks and customize the built-in functionality of TestStand. You learn advanced features of TestStand, such as using the TestStand API to dynamically manipulate the test system, creating custom step types and templates to support test developers, creating and customizing process models and user interfaces, and customizing data collection and report generation. An appendix provides additional information about customizing database logging. Learn how to create or customize the TestStand framework to meet any test system requirements.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Dynamically manipulate the test environment using the TestStand API
- Create custom step types and templates to support test developers
- Create and customize process models
- Create and customize user interfaces
- Customize report generation
- Design test system frameworks

Prerequisites

- TestStand 1: Test Development or equivalent experience
- Familiarity with LabVIEW, LabWindows/CVI, or C programming
- Recommended depending on your development application:
 - LabVIEW Core 1 and 2
 - LabWindows/CVI Core 1 and 2

NI Products Used

- TestStand Version 2012 or later
- LabVIEW Professional Development System Version 2012 or later
- (Optional) LabWindows/CVI Version 2012 or later
- Vision Development Module 2012 or later
- Vision Acquisition Software 2012 or later

Coursework Topics

Test Frameworks

Get an introduction to the concept of the test framework, which is defined as the components of an automated test system that are not specific to a particular type of unit under test.

TestStand API

Explore the TestStand API. Learn how it is organized using an object-oriented architecture; how to call it from TestStand; LabVIEW; and LabWindows/CVI; and how to use it to manipulate the test environment.

Custom Steps

Learn custom step types and compare them with step templates. Explore how custom step types influence the behavior of steps, such as modifying run-time behavior, modifying properties and results collection, creating dialog boxes to set step properties, modifying default step settings, and defining code templates. This lesson also discusses using step templates to customize steps and when a step template is the appropriate choice for customization.

Process Models

Review the process model, which controls how tests are configured and executed. Learn how to customize a process model to define configuration entry points, create custom execution entry points, include user prompts, modify data collection, and create custom reports.

User Interfaces

Examine the built-in TestStand user interfaces and how to use the TestStand User Interface Controls in LabVIEW and LabWindows/CVI to create custom user interfaces. This lesson also describes how user interface messages send asynchronous notifications between the process model and the user interface and the role of front-end callbacks in relation to user interfaces.

Design Considerations

Examine in detail recommended design decisions for creating an effective test framework, including the appropriate locations to implement client sequences, process models, custom step types, and user interfaces. Learn best practices for storing data, collecting results, handling errors, and deploying the test framework.

Suggested Next Courses

- Data Acquisition Using NI-DAQmx and LabVIEW
- LabVIEW Instrument Control
- Hardware Courses

Validate Your Expertise With an NI Certification

CTA | TestStand Architect

Prerequisite: CTD | TestStand Developer
Format: Multiple choice
Duration: 1 hour
Location: Pearson VUE testing centers or NI branch offices

Recertification Process

Recertification Interval: 4 years

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- **Develop Automated Production Test Systems**
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Data Acquisition Using LabVIEW NXG

Explore the fundamentals of data acquisition using sensors, NI data acquisition hardware, and LabVIEW software. Learn the basics of hardware selection, including resolution and sample rate, and the foundation of sensor connectivity, including grounding and wiring configurations. Use the NI-DAQmx driver to measure, generate, and synchronize data acquisition tasks and then program finite and continuous acquisitions. Discover best practices in hardware/software timing, triggering, and logging. Gain hands-on experience configuring and programming NI data acquisition hardware using NI-DAQmx and LabVIEW.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop integrated, high-performance data acquisition systems that produce accurate measurements
- Acquire data from sensors, such as thermocouples and strain gages, using NI DAQ hardware
- Apply advanced understanding of LabVIEW and the NI-DAQmx API to create applications
- Eliminate measurement errors due to aliasing and incorrect signal grounding
- Initiate measurements using hardware and software triggering
- Acquire and generate single-point and buffered analog waveforms
- Acquire and generate digital signals
- Use signal conditioning to improve the quality of acquired signals
- Synchronize multiple DAQ operations and devices

Prerequisites

- LabVIEW NXG Core 1 or equivalent LabVIEW experience

NI Products Used

- LabVIEW NXG 2.0
- NI-DAQmx 17.5 software
- cDAQ-9178 CompactDAQ Chassis
- C Series analog input, analog output, and digital I/O modules



Coursework Topics

Measuring Voltage or Current

Learn how to select and connect to data acquisition hardware based on resolution, range, and sensitivity; configure the NI-DAQmx analog input task with the correct sample rate; and validate an analog measurement.

Generating an Analog Signal

Understand the options for configuring the NI-DAQmx task to achieve an analog output generation, such as generating a current.

Generating or Reading a Digital Signal

Select the right DAQ hardware based on the type of digital signal you need to read or generate. Learn about connecting to sinking or sourcing devices and explore the characteristics of counter signals.

Exploring a Signal

Configure NI-DAQmx tasks and perform signal conditioning for bridge-based, IEPE, temperature, and counter measurements.

Programming With the NI-DAQmx API

Use the NI-DAQmx API to program applications that automate data communication between a DAQ device and computer.

Programming Multiple Channels

Explore ways you can create multichannel and multidevice tasks and when you can use them.

Triggering on a Specific Condition

Begin your data acquisition on a specific condition and understand how to use hardware sources as triggers.

Advanced Timing and Synchronization Methods

Learn about synchronization and use the appropriate method for synchronizing multiple DAQ tasks.

Logging Measurement Data to Disk

Log data to a TDMS file and store the data for analysis and post-processing.

System Considerations

Explore other aspects of building a data acquisition system, including hardware considerations like accuracy and bus and computer traits.

Suggested Next Courses

- LabVIEW Core 2
- LabVIEW Instrument Control
- DIAdem Basics
- Embedded Control and Monitoring Using LabVIEW

Suggested Badge Assessments

Taking Physical Measurements

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Data Acquisition Using NI-DAQmx and LabVIEW

Explore the fundamentals of data acquisition using sensors, NI data acquisition hardware, and LabVIEW software. Learn the basics of hardware selection, including resolution and sample rate, and the foundation of sensor connectivity, including grounding and wiring configurations. Use the NI-DAQmx driver to measure, generate, and synchronize data acquisition tasks and then program finite and continuous acquisitions. Discover best practices in hardware/software timing, triggering, and logging. Gain hands-on experience configuring and programming NI data acquisition hardware using NI-DAQmx and LabVIEW.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop integrated, high-performance data acquisition systems that produce accurate measurements
- Acquire data from sensors, such as thermocouples and strain gages, using NI DAQ hardware
- Apply advanced understanding of LabVIEW and the NI-DAQmx API to create applications
- Eliminate measurement errors due to aliasing and incorrect signal grounding
- Initiate measurements using hardware and software triggering
- Acquire and generate single-point and buffered analog waveforms
- Acquire and generate digital signals
- Use signal conditioning to improve the quality of acquired signals
- Synchronize multiple DAQ operations and devices

Prerequisites

- LabVIEW Core 1

NI Products Used

- LabVIEW Professional Development System
- NI-DAQmx
- CompactDAQ device

Coursework Topics

Measuring Voltage or Current

Learn how to select and connect to data acquisition hardware based on resolution, range, and sensitivity; configure the NI-DAQmx analog input task with the correct sample rate; and validate an analog measurement.

Generating an Analog Signal

Understand the options for configuring the NI-DAQmx task to achieve an analog output generation, such as generating a current.

Generating or Reading a Digital Signal

Select the right DAQ hardware based on the type of digital signal you need to read or generate. Learn about connecting to sinking or sourcing devices and explore the characteristics of counter signals.

Exploring a Signal

Configure NI-DAQmx tasks and perform signal conditioning for bridge-based, IEPE, temperature, and counter measurements.

Programming With the NI-DAQmx API

Use the NI-DAQmx API to program applications that automate data communication between a DAQ device and computer.

Programming Multiple Channels

Explore ways you can create multichannel and multidevice tasks and when you can use them.

Triggering on a Specific Condition

Begin your data acquisition on a specific condition and understand how to use hardware sources as triggers.

Advanced Timing and Synchronization Methods

Learn about synchronization and use the appropriate method for synchronizing multiple DAQ tasks.

Logging Measurement Data to Disk

Log data to a TDMS file and store the data for analysis and post-processing.

System Considerations

Explore other aspects of building a data acquisition system, including hardware considerations like accuracy and bus and computer traits.

Suggested Next Courses

- LabVIEW Core 2
- LabVIEW Instrument Control
- DIAdem Basics
- Embedded Control and Monitoring Using LabVIEW

Suggested Badge Assessments

Taking Physical Measurements

"The depth of material covered and the concepts highlighted are beyond comparison to other courses I've had."

Bruce Williams, Intertek

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

“NI's TestStand I is a great course. I highly recommend it.”

Donald Smith, Alfautomazione

Developing Test Programs Using TestStand

The Developing Test Programs Using TestStand course teaches you how to use TestStand to develop, analyze, debug, and deploy test applications. The course also introduces you to features in TestStand and the basics of using user interfaces, user privileges, reports, and process models to develop applications that match your test needs. Learn how to develop practical test applications in the TestStand environment and distribute them to test stations.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop practical test applications and distribute them to test stations
- Build test code to use in TestStand
- Use TestStand debugging tools
- Configure TestStand using the dialog boxes and customize user privileges
- Share data between TestStand and LabVIEW or LabWindows/CVI
- Understand how a test executive functions
- Log test results to a database and view with the Data Viewer

Prerequisites

- Familiarity with LabVIEW, LabWindows/CVI, or C programming
- Recommended depending on your development application
- LabVIEW Core 1
- LabWindows/CVI Core 1

NI Products Used

- TestStand Version 2012 or later
- LabVIEW Professional Development System Version 2012 or later
- LabWindows/CVI Version 2012 or later
- Vision Development Module 2012 or later
- Vision Acquisition Software Version 2012 or later

Coursework Topics

Introduction to Testing

Get an introduction to the fundamental concepts of tests and test systems, including the associated industry terminology, components of automated test systems, the roles of those who interact with and develop test systems, the test development process, and requirements tracking.

Introduction to TestStand

Learn the role of test management software, TestStand terminology, the components of TestStand (sequence editor, user interface, and TestStand Engine), and the TestStand execution architecture. This lesson also describes how to use the sequence editor to create, modify, execute, and debug sequences.

Analyzing Sequences

Explore the debugging and diagnostic tools for analyzing TestStand sequences, and tools to help you find information in sequence files and compare sequence files. Topics include debugging and diagnostics and the TestStand sequence analyzer.

Creating Sequences

Discover the steps involved in creating a test program. Learn how to create a new TestStand sequence, create steps, create and call code modules for steps, call subsequences, and configure sequence and sequence file properties.

Managing Data

Explore the data layout and how TestStand stores data in variables and containers. Learn to create expressions, define custom data types, and create three types of TestStand variables: locals, sequence file global variables, and station global variables. Also learn how to import and export properties for sequences.

Overriding Callbacks

Examine the concept of callback sequences, which can modify the behavior of the process model without modifying the process model itself, or allow the process model or sequence file to call certain sequences at various points during a test. Learn how and when to override process model callbacks and engine callbacks.

Configuring TestStand

Discover how to set preferences for TestStand stations, configure module adapters, edit search directories, and set report and database options.

Executing Tests in Parallel

Learn how to execute manufacturing tests using multiple test stations or parallel testing. Also discover how to use multithreading, parallel and batch process models, and synchronization to test multiple units in parallel.

Best Practices for Test Development

Review best practices for test development, including modular design techniques, appropriate data storage and communication mechanisms, and the proper use of setup and cleanup step groups. Learn how to abstract non-product-specific operations and include documentation in sequence files, sequences, and steps.

User Management

Get an introduction to user management in TestStand. Learn how to create user groups, add users, customize user privileges, and synchronize TestStand user management with external user management systems.

Deploying a Test System

Explore how to distribute a test system to other computers. Learn about considerations for deployment including deployed test system components, file paths, support files, and user interfaces. Also examine the deployment process and create a test system installer.

Suggested Next Courses

- Architecting Test Systems Using TestStand
- LabVIEW Core 2

Validate Your Expertise With an NI Certification

CTD | TestStand Developer

Prerequisite: None
Format: Multiple choice
Duration: 1 hour
Location: Pearson VUE testing centers or NI branch offices

Recertification Process

Recertification Interval: 3 years

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



DIAdem Basics

The DIAdem Basics course teaches you how to use the most important analysis, reporting, and data management features of the DIAdem environment. Discover how to import data sets, analyze them visually and computationally, and create publisher-quality reports. Also learn how to use DIAdem SCRIPT to automate each of these tasks.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Find the data you want to import using simple search terms and conditions
- Import ASCII files and load TDM files
- View and graphically investigate loaded data
- Execute channel-based calculations on loaded data
- Create reports to present the data, including graphs and tables
- Use DIAdem SCRIPT to automate the above steps
- Complete the most common tasks performed in DIAdem

Prerequisites

- Experience using Windows; programming experience is helpful but not essential

NI Products Used

- DIAdem

Coursework Topics

Introduction to DIAdem

Get an introduction to the DIAdem environment. This lesson examines the role, inherent operations, and main components of DIAdem.

Managing Data

Learn how to use data within the DIAdem environment. Explore how to search for data with the DataFinder and how data is organized in DIAdem. Topics include understanding the uses of the NAVIGATOR panel, learning about the components of the NAVIGATOR panel, and searching for data with the DataFinder.

Viewing Data

Use DIAdem to display channels as curves or tables; add text, pictures, or videos to the layout; and create, save, and load layouts. Topics include learning about the components of the VIEW panel, getting data into the VIEW panel, and loading and saving a DIAdem VIEW layout.

Reporting Data

Discover how to use DIAdem to display your data in graphs or in tables. Learn how to share results with others and collaborate on projects by communicating results clearly and concisely with a drag-and-drop environment tailored for creating engineering and scientific reports.

Analyzing Data

Explore how to use the ANALYSIS panel to perform interactive analysis using the DIAdem analysis libraries. Also learn how to analyze data using a wide range of mathematical functions.

Calculator and Calculation Manager

Use the Calculator to mathematically analyze data with your own formulas and use the Calculation Manager to create calculation templates, manage calculations, and repeat calculations.

Units in DIAdem

Learn how to execute calculations with units and which input help DIAdem provides. Topics include using units in DIAdem, understanding units definition and use, and editing the units catalog.

Automating Sequences

Learn how to automate recurring sequences and standard evaluations as scripts in DIAdem SCRIPT. Topics include understanding the uses of the SCRIPT panel, learning about the components of the SCRIPT panel, and automating DIAdem with the recording mode.

Executing Typical Tasks With DIAdem

Solve common tasks with DIAdem. Topics include loading non-DIAdem data, identifying and graphing peaks, finding events and trends, and working with 3D data.

Suggested Next Courses

- DIAdem Data Acquisition and Control
- DIAdem Advanced

Suggested Badge Assessments

DIAdem Data Exploration

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- **Develop Hardware-in-the-Loop Test Systems**
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- **Manage Data With DIAdem**
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

DIAdem Advanced

This course teaches you how to automate common tasks in the DIAdem panels with DIAdem's standard VBScript commands. You also learn how to customize the DIAdem environment by creating user-defined dialog boxes, user commands, and custom menus, which enable you to use DIAdem as the basis for complex applications.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Automate DIAdem using VBScript
- Use standard dialog boxes
- Design user-defined dialog boxes
- Create user-defined variables and commands
- Control the script sequence
- Define functions and procedures
- Optimize and structure scripts
- Programmatically handle files
- Import data with SQL queries in VBScript

Prerequisites

- DIAdem Basics or equivalent experience

NI Products Used

- DIAdem

Coursework Topics

Introduction to DIAdem SCRIPT

Examine the functions used for creating and editing scripts with the editor that is integrated in the DIAdem workspace. Topics include understanding the uses of the SCRIPT panel and learning about the components of the SCRIPT panel.

The VBS Functionality

Learn the most important aspects of the VBS functionality. Topics include understanding data types, variables, constants, and operators, and learning about condition instructions.

Working With the Debugger

Discover how to use the integrated debugger and breakpoints. Topics include determining error causes, working in debug mode, and minimizing errors in scripts.

Creating and Editing Scripts

Get an introduction to the technique of creating a script with the recording mode and learn techniques to edit the script to customize the behavior.

Variables and Commands in Scripts

Examine the different types of DIAdem variables, their value ranges, and the context in which you use them. Also learn about DIAdem commands and how to define your own commands.

Dialog Boxes in Scripts

See how to create dialog boxes to set function parameters and to specify settings to select properties and enter data. Topics include understanding the components of the Dialog Editor, creating user-defined dialog boxes, and using VBS in user-defined dialog boxes.

Working With DIAdem Panels

Learn how to access various DIAdem panels using a script. Topics include working with DIAdem REPORT layouts, displaying data with DIAdem VIEW, and accessing internal data.

Customizing DIAdem

Explore how to customize DIAdem for special applications. Topics include customizing the DIAdem environment, launching DIAdem with parameters, and working with the Bar Manager.

Advanced Script Options

This lesson describes a class definition and how to use objects derived from the class. It also highlights database access with ActiveX data objects.

Suggested Next Courses

- LabVIEW Core 1

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem**
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

DIAdem Data Acquisition and Control

The DIAdem Data Acquisition and Control (DAC) course teaches you how to acquire and visualize data using the DIAdem environment. Explore the DIAdem panels DAC and VISUAL, which are used for online tasks. In DIAdem DAC, create block diagrams to acquire measurement data, process the data online, and output the data. In DIAdem VISUAL, display the acquired and calculated measured values and enable a user to interact with the block diagram using operating controls. This course teaches you to acquire measurement data, process the data inline, and display the acquired and calculated measured values.

- C** Classroom: 1 day
- S** On-Site: 1 day
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

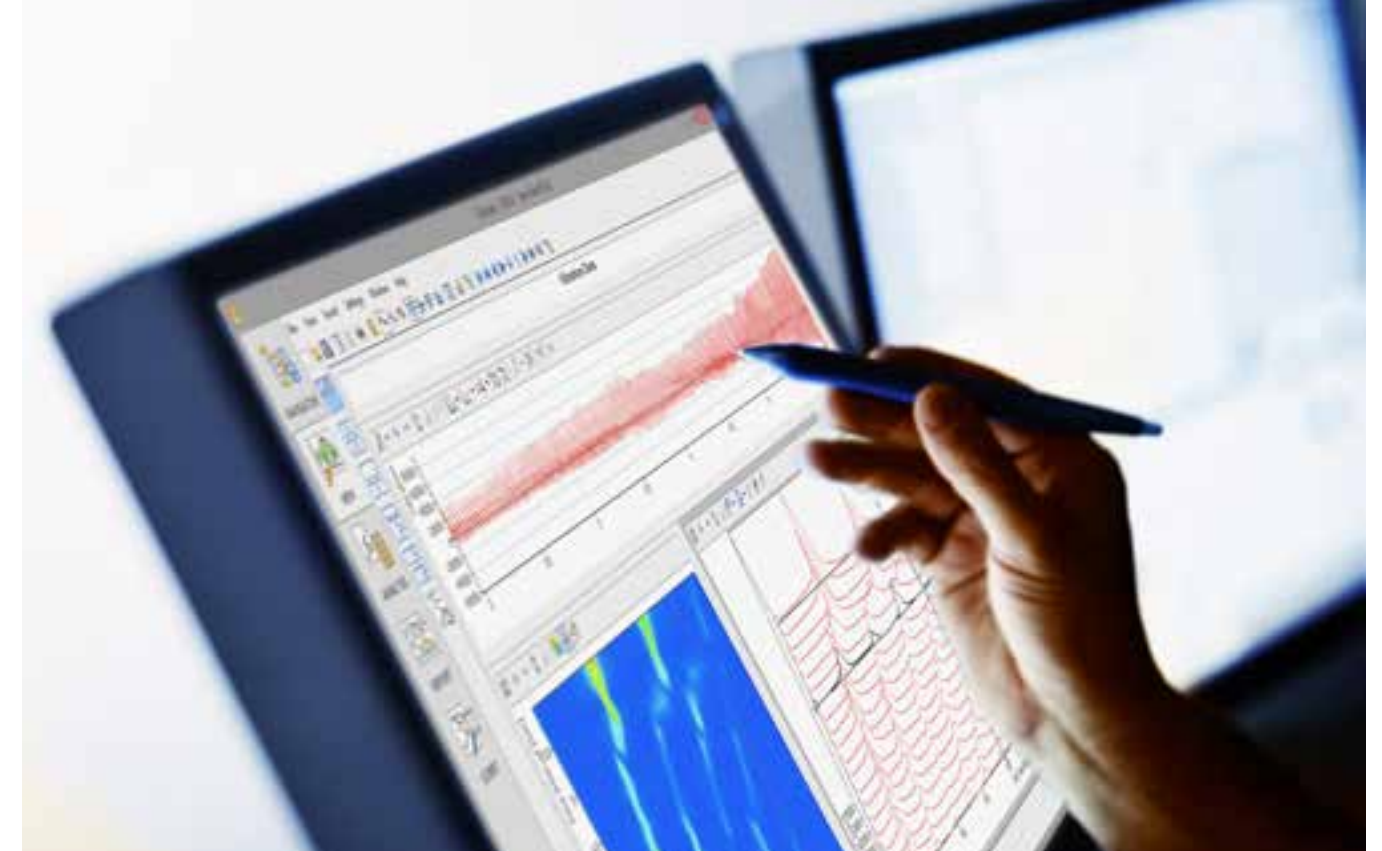
- Measure, control, and visualize your data
- Configure NI hardware for data acquisition
- Create user interfaces with charts, graphs, and buttons
- Display and log data
- Simulate processes and signals

Prerequisites

- Experience using Windows; programming experience is helpful but not essential
- DIAdem Basics or equivalent experience

NI Products Used

- DIAdem



Coursework Topics

Measuring, Controlling, and Visualizing Data

This lesson describes the DAC and VISUAL DIAdem panels, which you use for online tasks. In DIAdem DAC, you create block diagrams to acquire measurement data, process the data online, and output the data. In DIAdem VISUAL, you display the acquired and calculated measured values and enable the user to interact with the block diagram using operating controls.

Suggested Next Courses

- DIAdem Advanced

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem**
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Embedded Control and Monitoring Using LabVIEW

The Embedded Control and Monitoring Using LabVIEW course delivers hands-on training for designing, prototyping, and deploying a reliable embedded control and monitoring application. Learn how translate your embedded system requirements into a scalable software architecture, choose appropriate methods for interprocess and network-based communication, design your real-time application for reliability, and efficiently deploy and replicate your embedded system.

- C** Classroom: 5 days
- S** On-Site: 5 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Design, prototype, and deploy an embedded control and monitoring application
- Acquire and generate analog and digital signals, control timing, and implement signal processing on real-time and FPGA targets
- Implement functionality on the FPGA for maximum performance and reliability using the LabVIEW FPGA Module
- Implement additional functionality on the real-time target for logging, network communication, system health monitoring, and reliability using the LabVIEW Real-Time Module
- Implement a human machine interface (HMI) on the PC
- Communicate data between FPGA, real-time, and PC targets
- Debug, benchmark, and test your application
- Deploy your application to multiple systems

Prerequisites

- LabVIEW Core 1 and 2 or equivalent experience

NI Products Used

- LabVIEW Professional Development System
- LabVIEW FPGA Module
- LabVIEW Real-Time Module
- CompactRIO Controller
- Analog input, analog output, thermocouple input, and digital output modules

Coursework Topics

Introduction to Embedded Control and Monitoring

Get an introduction to embedded control and monitoring systems using LabVIEW. Learn about the FPGA, real-time, and HMI components of the system.

Configuring Your Hardware

Explore how to set up real-time hardware and software. Gain hands-on experience in configuring a CompactRIO real-time system.

Identifying Application Requirements

Examine a variety of considerations to make when designing and developing an embedded control and monitoring application.

Documenting Your Design

Learn how to use and create different types of diagrams to document your system design.

Accessing Your I/O in LabVIEW

Discover how to access the inputs and outputs of your real-time target in your application and learn to configure real-time targets through the LabVIEW project.

Programming Using LabVIEW FPGA

Explore how to program the FPGA using the LabVIEW FPGA Module. Gain a high-level understanding of how logic is implemented on the FPGA and how LabVIEW code is translated and compiled into FPGA hardware. After developing an FPGA VI, test, debug, compile, and then execute it on an FPGA target. Examine the different reports generated during compilation and learn techniques to optimize your code for size.

Using FPGA I/O and Timing

Learn how to add FPGA I/O to your LabVIEW project and access it on the block diagram using FPGA I/O nodes. Also explore how to set FPGA loop rates, add delays between events, and benchmark your FPGA code.

Signal Processing

Learn a variety of ways to process your signals in an FPGA VI. Topics include using fixed-point and single-precision floating-point data types as well as performing FPGA math and analysis.

Interprocess Communication in FPGA

Discover how to transfer data between multiple loops on your FPGA VI. Examine data-sharing methods including variables, memory items, register items, and FPGA FIFOs. Learn the benefits of each technique and when each should be used.

Communicating Between the FPGA and RT VIs

Learn how to transfer data between your FPGA VI and RT VI. Topics include programmatically communicating with the FPGA VI from the RT VI, deploying an FPGA VI, and transferring the latest data.

Optimizing FPGA Code

Explore how to use techniques to optimize for speed or FPGA size if necessary. Also examine pipelining.

Programming Using LabVIEW Real-Time

Learn how to design RT VIs and explore setting priorities and improving the determinism of an application.

Interprocess Communication in an RT VI

Discover how to transfer data between multiple loops on your RT VI and learn more about sharing data between deterministic and nondeterministic processes.

Communicating Between a Real-Time Target and a PC

Communicate between the real-time target and PC and explore communication techniques such as network-published shared variables, network streams, TCP, UDP, and reference libraries.

Managing Memory and Monitoring System Health

Examine techniques to manage memory properly and monitor the health of the embedded system.

Reliability

Explore techniques to improve the reliability of the real-time application and examine safe shutdown, specific and central error handling, and redundancy.

Debugging, Benchmarking, and Testing

Review debugging tools, benchmark performance and code duration, test a real-time application, and more.

Suggested Next Courses

- LabVIEW Core 3
- LabVIEW Machine Vision
- High-Throughput FPGA
- FlexRIO

Validate Your Expertise With an NI Certification

Certified LabVIEW Embedded Systems Developer (CLED)

Prerequisite: CLD or CLA
Format: 2-part exam, multiple choice and application development
Duration: 6 hours
Location: NI branch offices or on-site at your location

Recertification Process

Recertification Interval: 5 years

FlexRIO

The FlexRIO course teaches you to design high-performance systems using FlexRIO FPGA and adapter modules. Learn about FlexRIO hardware architecture and features, from the extremely high-performance I/O to the real-time and low-latency FPGA-based processing. This course is an extension of the High-Throughput LabVIEW FPGA course.

- C** Classroom: 1 day
- S** On-Site: 1 day
- V** Virtual: four 4-hour sessions
- O** Online: self-paced



Coursework Goals

- Understand the basic architecture and capabilities of FlexRIO FPGA modules, including the clocking architecture, PXI interface, DRAM performance, and adapter module card-edge connector
- Understand common adapter module architectures and how you can use them in different applications
- Optimize FlexRIO-based systems for throughput and latency performance, especially when using peer-to-peer (P2P) streaming

Prerequisites

- High-Throughput LabVIEW FPGA or equivalent experience
- Familiarity with high-performance LabVIEW FPGA programming

NI Products Used

- LabVIEW Professional Development System Version 2011 or later
- LabVIEW FPGA Module Version 2011 or later
- LabVIEW Digital Filter Design Toolkit
- NI-RIO device driver
- FlexRIO FPGA and adapter modules
- FlexRIO Instrument Development Library and example instrument drivers
- Peer-to-peer driver

Coursework Topics

Introduction to FlexRIO Hardware
Explore the FlexRIO hardware architecture, including general FPGA module features and architectures, FPGA module clocking options, and digital and analog adapter module architectures and clocking options.

Identifying and Analyzing System Throughput and Latency
Examine throughput and latency for real-time streaming systems, which require guaranteed minimum throughput and maximum latency. Topics include optimizing throughput and latency for single-FPGA systems, benchmarking systems for actual throughput and latency performance, and optimizing P2P throughput and latency for FPGA modules and P2P-enabled instruments.

Suggested Next Courses

- Managing Software Engineering in LabVIEW
- Object-Oriented Design and Programming in LabVIEW
- LabVIEW Performance

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

High-Throughput LabVIEW FPGA

The High-Throughput LabVIEW FPGA course teaches you the essential skills and processes to become proficient at designing high-performance applications with LabVIEW FPGA. Learn about FPGA hardware resources and how LabVIEW code maps to these resources. Review case studies, architectural considerations, and simulations to learn NI-recommended design flows when implementing large FPGA applications.

Note: NI offers two courses on the LabVIEW FPGA Module. The class most appropriate for you depends on the hardware and I/O rate of your application.

Note: LabVIEW FPGA focuses on programming practices for applications with I/O less than 5 MHz or using R Series, CompactRIO, or Single-Board RIO targets.

- C Classroom: 3 days
- S On-Site: 3 days
- V Virtual: four 4-hour sessions
- O Online: self-paced

Coursework Goals

- Understand how LabVIEW VIs are compiled for FPGA targets
- Develop applications on top of the NI-RIO device driver
- Develop high-performance DSP algorithms and digital designs in LabVIEW for FPGAs
- Integrate existing code (both LabVIEW and HDL) into LabVIEW FPGA applications
- Architect high-performance FPGA-based systems
- Follow recommended development and debugging processes for FPGA implementation

Prerequisites

- LabVIEW Core 1 or equivalent experience
- Familiarity with FPGA development using LabVIEW FPGA or otherwise is helpful but not required
- Familiarity with FPGA targets in the LabVIEW project and environment is helpful but not required

NI Products Used

- LabVIEW Professional Development System Version 2011 or later
- LabVIEW FPGA Module Version 2011 or later
- LabVIEW Digital Filter Design Toolkit
- NI-RIO device driver
- FlexRIO FPGA and adapter modules

Coursework Topics

Graphical Design in LabVIEW FPGA

Gain insight into FPGA hardware resources and how LabVIEW code maps to these resources in terms of size and propagation delay/loop rate.

Host Communication With LabVIEW FPGA Through NI-RIO

Examine the methods of communicating between host and FPGA VIs. Topics include memory-mapped register access and interrupts.

I/O in LabVIEW FPGA

Explore the many methods of connecting the LabVIEW FPGA VI to other logic on the FPGA as well as I/O outside the FPGA.

Crossing FPGA Clock Domains

Large applications often involve logic running at multiple rates. You must make specific considerations for exchanging data between these "clock domains." Examine use cases for multiple clock domains, global and local variables, and memory items.

LabVIEW FPGA Algorithm Design: DSP Case Study

To illustrate the NI-recommended design flow, this lesson presents a case study on DSP design, including recommended FPGA design flow, FIR filter introduction, and IP sources.

LabVIEW FPGA Algorithm Design: Digital Protocol Case Study

To illustrate the NI-recommended design flow, this lesson presents a case study on digital protocol implementation.

LabVIEW FPGA Architectures

You must make additional architectural considerations when building large FPGA applications. This lesson covers a design from concept to implementation.

Simulating and Debugging FPGA Designs With Third-Party Tools

Though most designs can be simulated with sufficient fidelity in LabVIEW, for completely bit-true, cycle-accurate simulation, some designs may require a third-party cycle-accurate simulator. Topics include an overview of cycle-accurate simulation and simulation in Mentor Graphics ModelSim and Xilinx ISim.

Suggested Next Courses

- FlexRIO

"Excellent course! All of my questions were answered, and the examples presented in the class were real-world problems I could apply to my own projects."

Ken Morgan, Westinghouse Electric Co.

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



HIL Fundamentals Using NI VeriStand

The HIL Fundamentals Using NI VeriStand course teaches you to use the out-of-the-box features of VeriStand to develop real-time test applications. This course introduces real-time test and walks you through building a VeriStand test system that includes real-time simulation, hardware I/O, and alarming. You also learn to create sophisticated real-time stimulus profiles to stimulate a unit under test and log its responses to a variety of real-world conditions.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop a real-time test system using VeriStand
- Create a run-time editable user interface for your real-time test system
- Create and modify stimulus profiles and real-time sequences using the Stimulus Profile Editor

Prerequisites

- Experience with Microsoft Windows

NI Products Used

- VeriStand 2017 or later
- Distributed System Manager 2017 or later
- PXIe-1071 chassis
- PXIe-8102 real-time controller
- PXIe-6341 data acquisition device
- CB-68LP connector blocks

Coursework Topics

Introduction to Real-Time Test

Explore the fundamental concepts of real-time test. Learn the terminology and roles associated with real-time test and explore different types of closed-loop control test, including model-in-the-loop and hardware-in-the-loop test.

Introduction to VeriStand

Discover the VeriStand environment, its components, and the architecture of an VeriStand system.

Configuring Hardware

Learn about the various types of hardware that VeriStand can interact with as part of your real-time test system. This lesson provides an overview of each type of hardware and describes how you can add each component to your VeriStand system.

Importing and Mapping to Models

Get an introduction to the concept of software modeling and learn how you can import, configure, and use models developed in a variety of environments within VeriStand.

Exploring Engine Features

Examine the features of the VeriStand Engine that you can use to develop a more robust application. Learn to create your own channels for storing data and performing calculations. Also learn how to trigger alarms and procedures that react to various system conditions and how to distribute your VeriStand system across one or more hosts and targets.

Configuring the Workspace

Review how you can monitor and modify your system from the Workspace window. Learn about the types of controls and indicators that you can place on your workspace and how you can use screens to organize them. Also learn about the different types of tools that you can launch from the workspace to monitor and configure your system as well as how you can configure user permissions for your system.

Creating Stimulus Profiles

Get an introduction to the Stimulus Profile Editor and learn how you can use it to stimulate the unit under test and log its responses.

Benchmarking and Improving Performance

Discover how you can measure the execution speed and CPU usage of your VeriStand system and learn techniques for improving each. Topics include benchmarking and improving performance.

Extending VeriStand

Explore how you can extend the functionality of VeriStand. Learn how to use VeriStand functionality from other applications, custom workspace tools and objects to extend the workspace, and custom devices to extend the VeriStand Engine.

Troubleshooting

Discover techniques to troubleshoot issues that might arise during your VeriStand application development.

Suggested Next Courses

- LabVIEW Core 1
- Embedded Control and Monitoring Using LabVIEW
- Developing Test Programs Using TestStand
- DIAdem Basics

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- **Develop Hardware-in-the-Loop Test Systems**
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

LabVIEW Communications 802.11 Application Framework

This course provides an introduction to the LabVIEW Communications 802.11 Application Framework, which features FPGA-based 802.11 PHY and MAC layer IP. Learn how the open and modifiable FPGA IP blocks are structured and how they talk to the host so you can modify the blocks to fit your application needs. This course also includes instruction on the FPGA programming environment for LabVIEW NXG to show how you can use the framework for Wi-Fi-based prototyping.

- C Classroom: 3 days
- V Virtual: four 4-hour sessions
- S On-Site: 3 days
- O Online: self-paced

Coursework Goals

- Explore which features of the 802.11 physical layer are implemented
- Learn how the code is implemented in the host environment
- Learn how the code is implemented in the FPGA environment

Prerequisites

- LabVIEW NXG Core 1
- LabVIEW NXG Core 2
- LabVIEW FPGA

NI Products Used

- LabVIEW Communications System Design Suite
- LabVIEW Communications 802.11 Application Framework

Coursework Topics

Introduction

Get an introduction to the LabVIEW Communications 802.11 Application Framework.

802.11 Features

Explore which features of the 802.11 physical layer are implemented.

Host Implementation

Learn how the code is implemented in the host environment.

FPGA Implementation

Learn how the code is implemented in the FPGA environment.

Suggested Next Courses

- LabVIEW Communications LTE Application Framework

LabVIEW Communications LTE Application Framework

This course provides an introduction to the LabVIEW Communications LTE Application Framework, which features FPGA-based LTE PHY and MAC layer IP. Learn how the open and modifiable FPGA IP blocks are structured and how they talk to the host so you can modify the blocks to fit your application needs. This course also includes instruction on the FPGA programming environment for LabVIEW NXG to show how you can use the framework for LTE-based prototyping.

- C Classroom: 3 days
- V Virtual: four 4-hour sessions
- S On-Site: 3 days
- O Online: self-paced

Coursework Goals

- Examine LTE features in the physical layer
- Use the LTE Application Framework as the basis for prototyping new algorithms and systems
- Understand design principles and implementation techniques

Prerequisites

- LabVIEW NXG Core 1
- LabVIEW NXG Core 2
- LabVIEW FPGA

NI Products Used

- LabVIEW Communications System Design Suite
- LabVIEW Communications LTE Application Framework

Coursework Topics

Introduction

Get an introduction to the LabVIEW Communications LTE Application Framework.

LTE Features

Explore which features of the LTE physical layer are implemented.

Working With the Code

Learn how to use the LTE Application Framework as the basis for prototyping new algorithms and systems.

Implementation

Examine design principles and implementation techniques.

Suggested Next Courses

- LabVIEW Communications 802.11 Application Framework

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



LabVIEW Connectivity

The LabVIEW Connectivity course teaches you to identify the components of integrated systems and implement networking technologies for your applications. You learn how to extend your application functionality and reduce development time by leveraging the capabilities of other applications using connectivity technologies such as DLLs, ActiveX, databases, and the Internet.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- **Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

- Identify the components, benefits, and use cases of different network communication options
- Design applications using different networking technologies and architectures
- Programmatically control LabVIEW VIs and applications using the VI Server
- Share data between LabVIEW and other applications over a network
- Create and deploy services using LabVIEW
- Use the LabVIEW Database Connectivity Toolkit to communicate with databases
- Use LabVIEW with ActiveX and .NET
- Use DLLs with LabVIEW
- Use UDP and TCP/IP VIs to communicate with other applications locally and over a network

Prerequisites

- LabVIEW Core 2 or equivalent experience

NI Products Used

- LabVIEW Professional Development System Version 2010 or later
- LabVIEW Database Connectivity Toolkit

Coursework Topics

Calling Shared Libraries in LabVIEW

Learn how to use LabVIEW to call code written in other languages. Also learn how to use the Call Library Function Node to call DLLs on Windows. Topics include a shared library overview, calling shared libraries, and using the import shared library wizard.

Using the VI Server

The VI Server is an object-oriented, platform-independent technology that provides programmatic access to LabVIEW VIs. Discover how to access the VI Server through block diagrams, ActiveX technology, and the TCP protocol to communicate with VIs and other application instances so you can programmatically control VIs and LabVIEW.

Using ActiveX and .NET Objects in LabVIEW

Learn how to extend your application functionality by accessing other Windows applications using .NET and ActiveX technologies and event programming. Topics include using ActiveX controls, servers, and documents in LabVIEW and calling the LabVIEW ActiveX server from other languages.

Connecting to Databases

Explore database terminology and demonstrate database programming in LabVIEW with topics that include complying with database standards, connecting to a database, and performing standard database operations.

Broadcasting Data and Serving Data to a Client

Explore how to use the User Datagram Protocol (UDP) as a means for communicating short packets of data to one or more recipients on a network and implementing the broadcast model in LabVIEW. Additionally, this lesson describes how to use TCP/IP to communicate and share data over single and interconnected networks. Learn to create diverse systems that share information using the client/server model of networked applications.

Using LabVIEW Web Services

Get an introduction to LabVIEW web services and discover how to use LabVIEW to provide a standard web-based interface for communication with a LabVIEW application. Learn how to use LabVIEW as both a client and server for your web service.

Suggested Next Courses

- LabVIEW Core 3
- LabVIEW Performance
- Object-Oriented Design and Programming in LabVIEW

Validate Your Expertise With an NI Certification

CLD | Certified LabVIEW Developer Certification

Prerequisite: None
 Format: Application development
 Duration: 4 hours
 Location: NI branch office, training center, online, or on-site at your location

Recertification Process
 Recertification Interval: 3 years

“I felt that the instructor was exceptional in his grasp of the subject matter and very effective in conveying concepts.”

— Ron Click, Robert Bosch Corporation



“LabVIEW Core 1 provided me with a useful understanding of how to create and maintain a project in LabVIEW.”

Ron Ryan, DME

LabVIEW Core 1

The first step in any LabVIEW learning path, LabVIEW Core 1 gives you the opportunity to explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format. In this course, learn to develop data acquisition, instrument control, data-logging, and measurement analysis applications. Also discover how to create applications using the state machine design pattern to acquire, process, display, and store real-world data.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

- Understand front panels, block diagrams, icons, and connector panels
- Create user interfaces with charts, graphs, and buttons
- Use programming structures and data types that exist in LabVIEW
- Use various editing and debugging techniques
- Create and save VIs for use as subVIs
- Display and log data
- Create applications that use data acquisition devices
- Create applications that use GPIB and serial port instruments
- Use the state machine design pattern in your applications

Prerequisites

- Experience with Microsoft Windows
- Experience writing algorithms in the form of flowcharts or block diagrams

NI Products Used

- LabVIEW Professional Development System
- NI data acquisition device
- IEEE 488.2 (GPIB) controller
- Instrument Simulator
- BNC-2120

Coursework Topics

Navigating LabVIEW

Explore the LabVIEW environment, including windows, menus, and tools, and learn to create LabVIEW projects. Use the LabVIEW front panel and block diagram and search for controls, VIs, and functions.

Creating Your First Application

Learn how to build a simple LabVIEW application that acquires, analyzes, and presents data. Topics include understanding the dataflow programming model, recognizing different data types, and using Express VIs to build a basic VI.

Troubleshooting and Debugging VIs

Learn debugging and error-checking techniques in LabVIEW and identify problems with block diagram organization and data passing within a block diagram.

Using Loops

Explore the different ways that you can iteratively execute LabVIEW code and techniques for managing loop execution. Topics include adding software timing to your code and sharing data between loop iterations.

Creating and Leveraging Data Structures

Examine data types that combine related data into a single structure for improved data access and analysis. Topics include creating and using array controls and indicators and using type definitions to improve reuse of data structures in applications.

Using Decision-Making Structures

Study LabVIEW structures that you can use to implement decision-making algorithms in your applications. Topics include creating and using Case and Event structures.

Modularity (SubVIs)

Discover modular programming in LabVIEW and learn how to build the icon and connector pane of a VI so that it can be used as a subVI. Topics include basics of modular programming and documenting your code.

Acquiring Measurements With Hardware

Examine the differences between NI DAQ systems and instrument control and how LabVIEW connects to hardware to create real-world measurements.

Accessing Files in LabVIEW

Learn basic file I/O concepts and how to access and modify resources in LabVIEW. Topics include available high-level and low-level file I/O functions.

Using Sequential and State Machine Programming

Explore common sequential LabVIEW design techniques and the state machine design pattern, including techniques for sequential programming and implementing a state machine design pattern.

Suggested Next Courses

- LabVIEW Core 2
- Data Acquisition Using NI-DAQmx and LabVIEW
- Embedded Control and Monitoring Using LabVIEW
- LabVIEW Instrument Control
- Other hardware courses

Suggested Badge Assessments

LabVIEW Programming

LabVIEW Core 2

The LabVIEW Core 2 course is an extension of LabVIEW Core 1 and teaches you to use common design patterns to successfully implement and distribute LabVIEW applications for research, engineering, and testing environments. Topics covered include programmatic control of your user interface, techniques to optimize reuse of existing code, use of file I/O functions, and tools to create executables and installers. This course directly links LabVIEW functionality to your application needs and provides a jump-start for application development.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Use local variables to modify front panel controls or stop parallel loops
- Apply common design patterns that use queues and events
- Programmatically control user interface objects
- Evaluate file I/O formats and use them in applications
- Modify existing code for improved usability
- Prepare, build, debug, and deploy stand-alone applications

Prerequisites

- Experience with Microsoft Windows
- LabVIEW Core 1 or equivalent experience

NI Products Used

- LabVIEW Professional Development System
- NI data acquisition device
- BNC-2120

Coursework Topics

Using Variables

Learn how to use local variables to modify front panel control values, stop parallel loops, and circumvent dataflow limitations. Topics include communicating between parallel loops and writing to controls and reading from indicators.

Communicating Data Between Parallel Loops

Explore asynchronous communication techniques for creating code that is UI-driven and synchronizes data between parallel loops. Topics include using queues to pass buffered data between loops and using notifiers to broadcast data to multiple loops.

Implementing Design Patterns

Get an introduction to design patterns and learn about the specific benefits and functionality of these design patterns. See how they can be used to reduce development time and learn two different categories of programming design patterns: single loop and multiple loops.

Controlling the User Interface

Study methods to control the attributes of front panel objects programmatically, such as temporarily disabling a control. Learn how to use the VI Server to access the properties and methods of front panel objects.

File I/O Techniques

Examine different file formats for collecting and storing data and how to select the appropriate file format for your applications. Practice implementing modular code that reads or writes measurement data and compares file formats.

Improving an Existing VI

Focus on methods to refactor inherited LabVIEW code and to maximize the reuse of existing code. Refactoring is the process of redesigning software to make it more readable and maintainable without altering its observable behavior.

Creating and Distributing Applications

See the process of creating stand-alone executables and installers for LabVIEW applications. Learn how to use the Application Builder in LabVIEW. Topics include preparing your files and creating build specifications.

Suggested Next Courses

- LabVIEW Core 3
- Data Acquisition Using NI-DAQmx and LabVIEW
- Embedded Control and Monitoring Using LabVIEW
- LabVIEW Instrument Control
- Other hardware courses

Suggested Badge Assessments

LabVIEW Application Development

Validate Your Expertise With an NI Certification

CLAD | Certified LabVIEW Associate Developer Certification

Prerequisite: None
Format: Multiple choice
Duration: 1 hour
Location: Pearson VUE testing centers or NI branch offices

Recertification Process
Recertification Interval: 2 years

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

“This course covers the important concepts for LabVIEW thoroughly and concisely.”

Stephen Cohen, Newton Photonics



LabVIEW Core 3

LabVIEW Core 3 introduces you to structured practices to design, implement, document, and test LabVIEW applications. Focus on developing hierarchical applications that are scalable, readable, and maintainable. The processes and techniques covered in this course help reduce development time and improve application stability. By incorporating these design practices early in your development, you avoid unnecessary application redesign, increase VI reuse, and minimize maintenance costs.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop an application in an agile manner
- Use Project Explorer Window tools to improve file organization and resolve conflicts
- Use the Queued Message Handler to develop a multiloop application
- Use notifiers for one-to-many communication between parallel loops
- Develop a customized user interface prototype for initial usability testing
- Handle errors locally or globally depending on the severity of the error

Prerequisites

- LabVIEW Core 1 and 2 or equivalent experience

NI Products Used

- LabVIEW Professional Development System

Coursework Topics

Developing Successful Applications

Learn various development techniques to create scalable, readable, and maintainable VIs. Discover the importance of following a software development process and how to develop code that satisfies project requirements.

Organizing the Project

Explore the tools and techniques for organizing and managing files in a LabVIEW project. Discover how to create and use project libraries, how to use various tools to learn more about each LabVIEW project file, and how to identify, resolve, and prevent cross-linking errors and file conflicts.

Creating an Application Architecture

Study techniques and programming practices for creating intuitive architectures for large applications. Learn the importance of testing your top-level architecture, the value of following established style guidelines, ways to implement user events and notifiers, and methods to use the Queued Message Handler project template to begin developing a multiloop application.

Customizing the User Interface

Improve the way you implement front panels in LabVIEW and learn how to use run-time menus, splitter bars, panes, subpanels, and tab controls to customize and extend your user interface. Also learn the value of creating a user interface prototype and techniques for improving the usability of your application.

Managing and Logging Errors

Examine several approaches to developing software that responds to different types of errors. Learn how to determine whether an error should be handled locally or globally and when you should log error data to disk for later analysis.

Creating Modular Code

Learn how to use modular code in a large application and guidelines for making large applications more maintainable. Explore several approaches for testing code modules and integrating them into your top-level application architecture.

Suggested Next Courses

- Object-Oriented Design and Programming in LabVIEW
- LabVIEW Performance
- LabVIEW Connectivity
- Managing Software Engineering in LabVIEW
- Advanced Architectures in LabVIEW
- Transitioning to LabVIEW NXG

Suggested Badge Assessments

LabVIEW Application Development

Validate Your Expertise With an NI Certification

CLD | Certified LabVIEW Developer Certification

Prerequisite: None
 Format: Application development
 Duration: 4 hours
 Location: NI branch office, training center, online, or on-site at your location

Recertification Process
 Recertification Interval: 3 years

“The LabVIEW FPGA online course gave me a good overview of FPGA techniques and as a result I will be able to immediately use FPGA.”

David Judd, Michelin

LabVIEW FPGA

The LabVIEW FPGA course prepares you to design, debug, and implement efficient, optimized applications using the LabVIEW FPGA Module and reconfigurable I/O (RIO) hardware. Learn how to compile and deploy your VIs to different types of NI targets, such as R Series multifunction RIO, CompactRIO, Single-Board RIO, and NI RIO instruments. You develop applications that teach you how to acquire digital and analog I/O, control loop timing, synchronize operations, implement signal processing, and pass data between your host VI and your FPGA target.

- C** Classroom: 3 days
- S** On-Site: 3 days by request
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Evaluate system architectures for LabVIEW FPGA and NI RIO hardware
- Select and configure RIO hardware
- Create and compile your LabVIEW FPGA VI and download it to RIO hardware
- Acquire and generate analog and digital signals, control timing, synchronize operations, and implement signal processing on the FPGA
- Communicate between the FPGA and a host
- Design and implement applications using the LabVIEW FPGA Module

Prerequisites

- LabVIEW Core 1 or equivalent experience

NI Products Used

- LabVIEW Professional Development System
- LabVIEW FPGA Module
- LabVIEW Real-Time Module
- PCIe-7852R R Series device (simulated)
- cRIO-9074 integrated controller and C Series modules

Note: This legacy course is available only online. Consider taking the **Embedded Control and Monitoring** course instead.

Coursework Topics

Introduction to LabVIEW FPGA

Examine FPGA technology and LabVIEW FPGA and learn the components of a LabVIEW FPGA system and types of applications that are well-suited for LabVIEW FPGA. Compare a modern LabVIEW FPGA system with a traditional LabVIEW FPGA system and a traditional measurement system through topics including components of a LabVIEW FPGA system, comparisons with NI-DAQmx, and examples of LabVIEW FPGA applications.

Getting Started With LabVIEW FPGA

Learn about the two major RIO architectures: FPGA on Windows and FPGA for real-time OSs. Also explore more about R Series devices and CompactRIO. Configure your RIO hardware in Measurement & Automation Explorer (MAX) and create a LabVIEW FPGA project.

Programming Using LabVIEW FPGA

Discover how to reconfigure an FPGA target using the LabVIEW FPGA Module. Gain a high-level understanding of how logic is implemented on the FPGA and how LabVIEW code is translated and compiled into FPGA hardware. After developing an FPGA VI, test, debug, compile, and then execute on an FPGA target. Examine different reports generated during compilation and learn techniques to optimize your code for size.

Using FPGA I/O

Learn how to add FPGA I/O to your LabVIEW project and use it on the block diagram. Also examine the differences between performing I/O on an R Series device and on a CompactRIO chassis. Using I/O nodes, learn how to access both analog and digital data.

Timing an FPGA VI

Examine how to set FPGA loop rates, add delays between events, and benchmark your FPGA code. Topics include timing Express VIs, implementing loop execution rates, and creating delays between events.

Executing Code in Single-Cycle Timed Loops

Study how to improve the performance of your FPGA VI by using the single-cycle Timed Loop (SCTL), which executes at the rate of selectable FPGA clocks. Topics include dataflow in FPGAs, SCTLs, troubleshooting SCTL errors, and using FPGA clocks.

Signal Processing

Learn a variety of ways to process your signals in an FPGA VI with topics such as fixed-point data types, single-precision floating-point data types, and third-party IP integration.

Sharing Data on an FPGA

Discover how to transfer data between multiple loops on your FPGA VI and examine data-sharing methods including variables, memory items, register items, and FPGA FIFOs. Learn the benefits of each technique and when each should be used.

Synchronizing FPGA Loops and I/O

Learn how to synchronize loops and I/O in your FPGA VI. Topics include synchronizing FPGA loops and C series I/O modules.

Modular Programming

Discover how to most efficiently use subVIs in your FPGA application. Learn when to set your VIs as reentrant or non-reentrant, depending on your FPGA needs. Also explore FPGA controls so that you can reference FIFOs, memory items, register items, FPGA clocks, and I/O nodes in subVIs.

Communicating Between the FPGA and Host

Learn how to interface with your FPGA VI from a Windows host or real-time host. Create host VIs to control and pass data between your FPGA and host system. Topics include programmatically communicating with the FPGA from the host, deploying an FPGA VI, transferring the latest data, and synchronizing the host VI and FPGA VI.

Suggested Next Courses

- LabVIEW Core 2
- Embedded Control and Monitoring
- LabVIEW Real-Time 1 and 2

“I was able to improve my programming skills; my code became more robust and more reusable. I became faster at programming and my development time went down.”

Thomas Bress, CLA

LabVIEW Instrument Control

The LabVIEW Instrument Control course prepares you to programmatically control instruments using LabVIEW. The benefits of instrument control include automated processes, time savings, and ease of use. During the course, examine real-world industry examples and learn to use VISA, a single interface to configure and control GPIB, Ethernet, serial, and VXI instruments. Also use, modify, and build a LabVIEW Plug and Play instrument driver, which is a set of commands contained in a modular API for an individual instrument. After you complete the course, you can quickly develop integrated, high-performance instrument control applications that produce accurate measurements.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Use LabVIEW to communicate with instruments
- Program with VISA, VISA properties, and VISA events
- Parse data
- Use instrument drivers
- Modify an existing instrument driver
- Design, develop, test, and deploy a new instrument driver
- Create an effective and efficient instrument control application

Prerequisites

- Experience with Microsoft Windows
- LabVIEW Core 1 or equivalent experience

NI Products Used

- LabVIEW Professional Development System Version 8.5 or later
- IEEE 488.2 (GPIB) board
- Instrument Simulator

Coursework Topics

Industry Applications

Learn the benefits of instrument control. Topics include an introduction to instrument control and its benefits.

Communicating With Instruments

Explore how you can use LabVIEW to communicate with instruments and learn how to use NI-VISA, I/O Trace, SCPI commands, and the Instrument I/O Assistant to perform instrument I/O.

Using Instrument Drivers

Get an introduction to LabVIEW Plug and Play instrument drivers. Learn when to use instrument drivers, how to install instrument drivers, and how to program with instrument drivers.

Modifying an Existing Instrument Driver

If an instrument driver is lacking a feature you need, learn how and when to modify it. This lesson describes how to explore the structure of an instrument driver and steps to modify an instrument driver.

Creating a New Instrument Driver

Sometimes you have an instrument that does not have an instrument driver. Discover how to create a new instrument driver, design an instrument driver structure, and develop, test, and deploy an instrument driver.

Developing an Application

Examine how to develop effective and efficient instrument control applications. Develop a real-world, automated application and explore how to apply common, effective LabVIEW architectures for instrument applications.

Connecting to Databases

Examine database terminology and programming in LabVIEW. Topics include database standards, standard database operations, and structured query language.

Broadcasting Data and Serving Data to a Client

Discover how to use the User Datagram Protocol (UDP) as a means for communicating short packets of data to one or more recipients on a network and implementing the broadcast model in LabVIEW. Additionally, learn how to use TCP/IP to communicate and share data over single and interconnected networks and create diverse systems that share information using the Client/Server model of networked applications.

Using LabVIEW Web Services

Get an introduction to LabVIEW web services and learn how to use LabVIEW to provide a standard web-based interface for communication with a LabVIEW application. Also learn how to use LabVIEW as both a client and server for your web service.

Suggested Next Courses

- LabVIEW Core 2
- LabVIEW Core 3
- Data Acquisition Using NI-DAQmx and LabVIEW
- Modular Instruments 2: Digital Multimeters

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum**
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



LabVIEW Machine Vision

NI products give you the flexibility to address machine vision needs in your research, test and measurement, and industrial automated vision applications. This course teaches you to understand the fundamentals of machine vision, use and select the hardware components of a machine vision system, and implement machine vision tasks and inspections using LabVIEW.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Understand the fundamentals of selecting and setting up the appropriate machine vision hardware components
- Develop a machine vision application in LabVIEW
- Acquire and display images in LabVIEW using machine vision hardware
- Use several image processing functions to process, analyze, and inspect images

Prerequisites

- LabVIEW Core 1 or equivalent experience

NI Products Used

- LabVIEW Professional Development System
- Vision Acquisition Software
- Vision Development Module
- Machine vision development kit with camera, lens, lighting, and optics

Coursework Topics

Introduction to Machine Vision and Image Acquisition

Receive an introduction to machine vision and an overview of the components in a complete machine vision system.

Introduction to Lighting, Camera, and Optics

Learn about the fundamentals and purpose of lighting, cameras, lenses, and optical accessories. Also explore how to select and use each of these components.

Machine Vision Solution Strategies

Study various hardware options when building a machine vision system. Learn about a variety of options for lighting, camera, and optics. Explore the NI solutions for machine vision. Then learn how to choose and design the hardware portion of your vision system.

Acquiring and Displaying Images in LabVIEW

Discover how to acquire and display images in LabVIEW.

Getting Measurement-Ready Images

See how to prepare an acquired image for measurements in LabVIEW. You learn how to analyze the image using a histogram, improve the contrast using a lookup table, and enhance its features using gray morphology and filters.

Performing Particle Analysis

Discover how to perform particle analysis on an image. Learn how to create a binary image using a threshold, prepare the binary image using morphology and particle filters, and obtain several particle measurements.

Machine Vision Functions

Explore how to perform various machine vision functions on an image. Also learn how to set up a coordinate system using edge detection or pattern matching. Then see how to make a variety of distance and analytic geometry measurements on objects in your image.

Calibrating Images

Learn how to get measurements in real-world units by calibrating the image.

Machine Vision Inspections

Explore how to perform a variety of machine vision inspections. Learn how to make measurements for metrology, inspect for presence or absence, inspect for defects, identify parts using bar codes and optical character recognition (OCR), use optical character verification (OCV), process color images, and more.

Considerations for Creating a Successful Vision Solution

Examine several considerations for requirements gathering and analysis when creating a vision solution. This lesson also walks through a process for designing, developing, debugging, and deploying a machine vision system solution.

Suggested Next Courses

- LabVIEW Core 2
- LabVIEW Core 3

- Develop LabVIEW Applications
- **Design Smart Machines or Industrial Equipment**
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- **Acquire Images With Machine Vision**
- Design Wireless Communications Systems

LabVIEW NXG Core 1: Acquire, Analyze, Visualize

Take advantage of this first step on the LabVIEW NXG learning path to explore the LabVIEW NXG environment, interactive analysis, dataflow programming, and common development techniques in a hands-on format. Learn how to develop data acquisition, instrument control, data-logging, and measurement analysis applications. Create applications using the state machine design pattern to acquire, analyze, process, visualize, and store real-world data.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Interactively acquire and analyze data from NI hardware (NI DAQ devices) and non-NI instruments (GPIB instruments)
- Create and program a LabVIEW NXG application that acquires, analyzes, and visualizes data
- Create user interfaces with charts, graphs, and buttons
- Use programming structures, data types, and the analysis and signal processing algorithms in LabVIEW NXG
- Debug and troubleshoot applications
- Work with sets of single-channel and multi-channel data
- Log data to file
- Use best programming practices for code reuse and readability
- Implement a sequencer using a state machine design pattern

Prerequisites

- Experience with Microsoft Windows
- Experience writing algorithms in the form of flowcharts or block diagrams

NI Products Used

- LabVIEW NXG
- NI data acquisition device
- IEEE 488.2 (GPIB) controller

Coursework Topics

Introduction to LabVIEW NXG

Get an introduction to LabVIEW NXG and common types of LabVIEW NXG applications.

First Measurement (NI DAQ Device)

Interactively acquire, analyze, and visualize data from an NI DAQ device. Connect your NI DAQ device, validate the data, and troubleshoot unexpected I/O results.

First Measurement (Non-NI Instrument)

Interactively acquire, analyze, and visualize data from a non-NI instrument.

Exploring an Existing Application

Explore an existing LabVIEW NXG project and application and predict the behavior of the application based on the dataflow programming model.

Creating Your First Application

Create a simple LabVIEW NXG application that acquires, analyzes, and visualizes data from NI DAQ devices and non-NI instruments.

Debugging and Troubleshooting

Debug and troubleshoot a LabVIEW NXG application. Correct a broken VI, use debugging techniques and tools, capture data from a VI panel, and manage errors.

Executing Code Repeatedly Using Loops

Use While Loops and For Loops to execute code repeatedly. Learn how to access data from previous loop iterations.

Working With Sets of Data

Use and process array and waveform data types and multichannel acquisition data.

Executing Code Based on a Condition (Case Structure)

Create and configure Case Structures to execute code based on a condition.

Writing and Reading Data to File

Access and modify file resources in LabVIEW NXG. Write data to a text file, programmatically create file and folder paths, and analyze data in a text file.

Reusing Code (SubVIs)

Use modular programming and well-documented subVIs to reuse code and improve code readability.

Grouping Data of Mixed Data Types (Clusters)

Improve data organization and code readability by grouping data of mixed data types into a cluster. Learn to disassemble and modify clusters and visualize clusters and cluster arrays.

Propagating Data Type Changes Using G Types

Automatically propagate data type changes made to a G type to all instances of that G type.

Implementing a Sequencer (State Machine)

Explore common sequential LabVIEW NXG design techniques and the state machine design pattern.

Suggested Next Courses

- LabVIEW NXG Core 2: Create User-Driven Applications

Suggested Badge Assessments

LabVIEW NXG Programming

- Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators**
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



LabVIEW NXG Core 2: Create User-Driven Applications

Building on LabVIEW NXG Core 1, the LabVIEW NXG Core 2 course teaches you to create user-driven applications. Learn to implement parallel loops, programmatically respond to user interface events, use proven design patterns, manage configuration settings in configuration files, and develop an error-handling strategy for your application. This course directly links LabVIEW NXG functionality to your application needs and provides a jump-start for application development.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Implement multiple parallel loops and transfer data between the loops
- Create an application that responds to user interface events
- Implement robust user-driven applications using proven design patterns
- Manage configuration settings for your application
- Develop an error-handling strategy for your application
- Generate execution logs in your application

Prerequisites

- Experience with Microsoft Windows
- LabVIEW NXG Core 1 or equivalent experience

NI Products Used

- LabVIEW NXG
- NI data acquisition device

Coursework Topics

Implementing Parallel Loops

Integrate multiple parallel loops into a single application and transfer data between loops using duplicate terminals or queues. Learn to stop parallel loops and identify and avoid race conditions.

Creating an Event-Driven User Interface

Create an application that responds to user interface events using event-driven design patterns such as the user interface event handler, event-driven state machine, producer-consumer (events), and the Queued Message Handler design pattern.

Controlling the User Interface

Modify the attributes of front panel objects programmatically, such as temporarily disabling a control. Use VI Server to access the properties of front panel objects

Managing Configuration Settings in an Application

Store and load configuration settings for your application from a delimited or INI configuration file.

Developing an Error-Handling Strategy

Develop a strategy for responding to errors. Analyze code for potential error-handling, test your error handling logic, and implement various error-handling strategies.

Distributing Code:

Create stand-alone executables and package installers for LabVIEW NXG applications.

Suggested Badge Assessments

LabVIEW NXG Application Development

Validate Your Expertise With an NI Certification

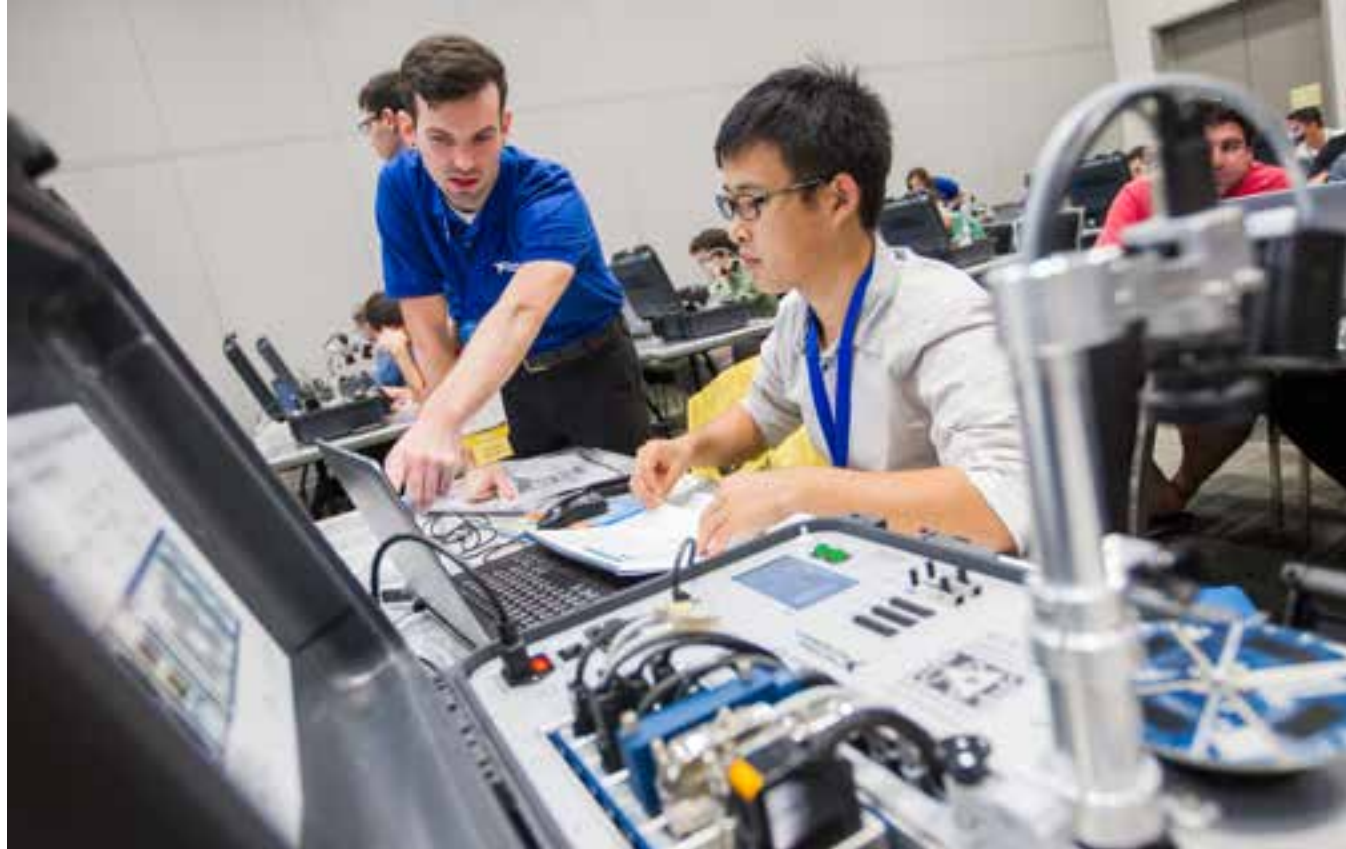
CLAD | Certified LabVIEW Associate Developer Certification

Prerequisite: None
Format: Multiple choice
Duration: 1 hour
Location: Pearson VUE testing centers or NI branch offices

Recertification Process

Recertification Interval: 2 years

- Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators**
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



LabVIEW Performance

The LabVIEW Performance course teaches techniques for improving the performance of your LabVIEW applications. Given application requirements, learn how to design an application to avoid common performance issues. Also choose the appropriate tools to measure application performance. After identifying performance issues, learn to modify code to improve memory usage and/or execution speed.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Describe how software and hardware interact and contribute to performance
- Design applications for optimal performance based on application requirements
- Select the appropriate tools to monitor application performance
- Identify performance issues in an existing application
- Modify existing applications to improve memory usage
- Modify existing applications to improve execution speed

Prerequisites

- LabVIEW Core 2 or equivalent experience

NI Products Used

- LabVIEW Professional Development System Version 2010 or later
- VI Analyzer Toolkit
- LabVIEW Desktop Execution Trace Toolkit

Defining Performance

Examine the basics of performance in relation to computer hardware and software. Also learn computer architecture basics and how system components affect performance. Topics include an overview of performance, computer resources, and resource management.

Designing Your Application

See how to design an application to avoid resource bottlenecks. Topics include managing CPU, memory, disk memory, threads, and I/O.

Measuring Performance

Examine various tools for measuring application and system performance. Then select the appropriate tool for measuring and identifying performance issues. Topics include bottleneck identification, performance measurement, the VI profiler, performance benchmarking, and the LabVIEW Desktop Execution Trace Toolkit.

Optimizing for Memory

Learn how to modify existing applications to improve memory usage. Also explore topics including memory buffers, memory reallocation, and memory cleanup.

Optimizing for Execution Speed

Discover how to modify existing applications to improve execution speed. Topics include UI updating, Event Structure and UI events, parallel code, and code simplification.

Tips, Tricks, and Other Techniques

Learn various tips, tricks, and techniques to help create your architecture. Topics include variant attributes, callback VIs with user events, VI scripting, and drop-in VIs.

Error Handling

Explore how to design a comprehensive error-handling strategy that can classify errors and take specific and general actions for errors. Topics include specific error-handling techniques that respond to specific errors, central error-handling techniques that respond to classes of errors, and an example LabVIEW error-handling reference library.

Additional Resources

Discover where to find additional code and ideas including through NI community groups and other forums.

Suggested Next Courses

- LabVIEW Core 3
- LabVIEW Connectivity
- Object-Oriented Design and Programming in LabVIEW
- Managing Software Engineering in LabVIEW
- Advanced Architectures in LabVIEW

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

“The LabVIEW Real-Time course gave me a good understanding of what is possible with real-time systems with excellent and to-the-point examples.”

Ajeet Singh, DeVry University

LabVIEW Real-Time 1

The LabVIEW Real-Time 1 course delivers hands-on training for prototyping deterministic measurement and control systems. At the end of the course, you will be able to design, develop, and prototype a real-time application that handles communication between the RT target and a host computer using NI-recommended methods and LabVIEW Real-Time.

- C Classroom: 3 days
- S On-Site: 3 days by request
- V Virtual: four 4-hour sessions
- O Online: self-paced

Coursework Goals

- Determine if a real-time solution is appropriate for a given problem
- Choose and configure the real-time target hardware for a given real-time application
- Implement a deterministic application
- Understand how to reduce the jitter in a real-time application
- Communicate between a host computer and real-time target using network communication

Prerequisites

- LabVIEW Core 1 or equivalent experience

NI Products Used

- LabVIEW Professional Development System
- LabVIEW Real-Time Module
- CompactRIO integrated controller
- CompactRIO thermocouple input module
- CompactRIO sourcing digital output module

Note: This legacy course is available only online. Consider taking the **Embedded Control and Monitoring** course instead.

Coursework Topics

Introduction to Real Time

Start learning the concepts of real-time development. Discover how to determine whether an application requires a real-time operating system or a general operating system. Also learn about the hardware architecture of a LabVIEW Real-Time system.

Configuring Your Hardware

Examine how to set up real-time hardware and software. Receive hands-on experience in configuring a CompactRIO real-time system. Topics include an overview of hardware setup and installation, network setting configurations for real-time targets and host computers, and real-time target configurations in Measurement & Automation Explorer.

Accessing Your I/O in LabVIEW

Learn how to access the inputs and outputs of your real-time target in your application. Topics include configuring real-time targets through the LabVIEW project and accessing I/O using driver APIs or the Scan Engine.

Real-Time Architecture

Explore how to design a real-time application. Learn about the general architecture of a real-time application, multithreading, priorities, and methods to improve the determinism of an application.

Interprocess Communication

Learn how to share data between multiple processes on your real-time target. Topics include sharing data locally on real-time targets, sharing data between deterministic and nondeterministic processes, and sharing data between nondeterministic processes.

Communicating Between Real-Time Target and Host

Implement the network communication portion of a real-time application. Learn about communicating with an application on a real-time target from the host computer. Also learn about various communication techniques such as front panel communication, network-published shared variables, network streams, TCP, and UDP.

Verifying Your Application

Discover methods for verifying the application behavior, performance, and memory usage of your application. Apply these methods to a real-time application. Topics include standard debugging techniques and performance memory usage verification.

Introduction to Deployment

Explore how to deploy a real-time application. Topics include creating a build specification and communicating with deployed applications.

Suggested Next Courses

- Embedded Control and Monitoring
- LabVIEW Real-Time 2
- LabVIEW FPGA
- LabVIEW Core 2
- LabVIEW Core 3

LabVIEW Real-Time 2

The LabVIEW Real-Time 2: Architecting Embedded Systems course delivers hands-on training for designing scalable, maintainable, and reliable embedded applications. Learn how to translate your embedded system requirements into a scalable software architecture, choose appropriate methods for interprocess and network-based communication, design your real-time application for reliability, and efficiently deploy and replicate your real-time systems.

C Classroom: 3 days

S On-Site: 3 days

V Virtual: four 4-hour sessions

O Online: self-paced

Coursework Goals

- Identify requirements and design a real-time application
- Document your system design using diagrams
- Implement the most appropriate method of sharing data locally on the real-time target between multiple processes
- Implement the most appropriate method of communicating between real-time targets and host computers over the network
- Properly manage memory and monitor the health of your system
- Implement a variety of techniques to increase the reliability of a real-time application
- Debug, benchmark, and test your real-time application
- Deploy your application to multiple systems

Prerequisites

- LabVIEW Real-Time 1 and LabVIEW Core 2 or equivalent experience

NI Products Used

- LabVIEW Professional Development System
- LabVIEW Real-Time Module
- Real-Time Execution Trace Toolkit
- CompactRIO integrated controller
- CompactRIO thermocouple input module
- CompactRIO sourcing digital output module

Note: This legacy course is available only online. Consider taking the **Embedded Control and Monitoring** course instead.

Coursework Topics

Real-Time Application Design Flow

Explore a variety of considerations to make when designing and developing a real-time application. Topics include analyzing your real-time application, real-time target considerations, and host considerations.

Documenting Your Design

See how to use and create different types of diagrams to document your system design. Topics include an overview of diagrams, communication diagram creation, and typical real-time system diagrams.

Real-Time Processes and Interprocess Communication

Examine different methods to share data between processes on the real-time target. Topics include comparing shared variables, real-time FIFO functions, and queues and using functional global variables for encapsulation and current value tables.

Network Communication

Study a variety of methods to communicate between real-time targets and host computers in different scenarios. Topics include selecting an appropriate network communication method and implementing a variety of network communication methods to monitor the latest values, stream data, and send messages in different scenarios.

Managing Memory and Monitoring System Health

Learn how to manage memory properly and monitor the health of a real-time target and application. Topics include impacts of memory usage, memory management, and system monitoring.

Reliability

Explore techniques to improve the reliability of a real-time application. Topics include safe states and shutdown, specific and central error handling, watchdog implementation, and redundancy.

Debugging, Benchmarking, and Testing

Examine methods for debugging, benchmarking, and testing your real-time application. Topics include debugging tools, benchmarking performance and code duration, and examining detailed traces of application behavior using the Real-Time Execution Trace Toolkit.

Deployment and Replication

Discover how to deploy and replicate a real-time application to multiple targets. Topics include target imaging and system replication.

Suggested Next Courses

- Embedded Control and Monitoring
- LabVIEW FPGA
- LabVIEW Core 3

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



LabWindows/CVI Core 1

The LabWindows/CVI Core 1 course prepares you to create applications using LabWindows/CVI. Discover how to create basic solutions using many of the built-in features of LabWindows/CVI and develop applications using the LabWindows/CVI programming environment. With the hands-on format, you can apply the skills you learned in the course to your applications.

- C** Classroom: 3 days
- V** Virtual: four 4-hour sessions
- S** On-Site: 3 days
- O** Online: self-paced

Coursework Goals

- Develop applications within the LabWindows/CVI environment
- Create professional user interfaces
- Understand the fundamentals of event-driven programming
- Create and use an instrument driver
- Develop and distribute stand-alone LabWindows/CVI applications
- Configure and use data acquisition within LabWindows/CVI
- Control a GPIB and serial instrument

Prerequisites

- Experience with C programming

NI Products Used

- LabWindows/CVI Professional Development System Version 2010
- NI-DAQmx 9.1.5
- NI-488.2 2.73 (GPIB) board
- Instrument Simulator
- BNC-2120

Coursework Topics

Introduction to LabWindows/CVI

Learn how to navigate within the integrated environment of LabWindows/CVI to develop, compile, and debug C code. Topics include creating function panels and accessing help and examples for function panels, using interactive execution to test code functionality, and building and linking your projects within the integrated environment.

Graphical User Interface

Discover how to design and build a graphical user interface. Get an introduction to the components of a user interface and how you can use the UI to control your code with event-driven programming. Also learn how easy it is to use LabWindows/CVI to create “skeleton code” that you can modify for your own application.

Advanced User Interface Techniques

Study how to create an architecture that provides a modular, scalable, and extensible user interface while exploring topics such as subpanels and XControls.

Introduction to Object-Oriented Programming in LabVIEW

See how using the encapsulation and inheritance capabilities of LabVIEW object-oriented programming can create scalable and extensible software architectures.

Plug-In Architectures

Explore how to create a plug-in architecture that you can use to add features without changing your main code. Topics include plug-in architectures using the VI Server and LabVIEW object-oriented programming, and the creation of subVIs from an existing VI.

Tips, Tricks, and Other Techniques

Learn various tips, tricks, and techniques to help create your architecture. Topics include variant attributes, callback VIs with user events, VI scripting, and drop-in VIs.

Error Handling

Examine how to design a comprehensive error-handling strategy that can classify errors and take specific and general actions for errors.

Additional Resources

Learn where to find additional code and ideas. Topics include NI community groups and other relevant forums.

Suggested Next Courses

- LabWindows/CVI Core 2
- Developing Test Programs Using LabVIEW

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- **Develop Automated Production Test Systems**
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- **Semiconductor Test System Curriculum**
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

LabWindows/CVI Core 2

A follow-up to the LabWindows/CVI Core 1 course, the LabWindows/CVI Core 2 course is ideal for new and intermediate users. After attending the course, you can design applications that use network communication, DLLs, and ActiveX. Discover how to create powerful user interfaces using menus and toolbars, and how to take full advantage of the PC with your application by learning how to program a multithreaded application.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Create toolbars and menus for user interfaces
- Use ActiveX within the LabWindows/CVI environment
- Compile your modules as DLLs you can use with other applications
- Use LabWindows/CVI to access DLLs
- Develop efficient multithreaded applications
- Develop applications that can communicate with the Internet
- Decrease development time using debugging features
- Optimize execution speed with the Execution Profiler Toolkit

Prerequisites

- LabWindows/CVI Core 1 or equivalent experience

NI Products Used

- LabWindows/CVI Full Development System Version 2010
- LabWindows/CVI Execution Profiler Toolkit
- GPIB interface and cable (optional)
- Instrument Simulator (optional)

Coursework Topics

Review of Core 1 Concepts

Review important concepts from the LabWindows/CVI Core 1 course. Topics include operation of the callback function model in LabWindows/CVI, operating system messaging and LabWindows/CVI, and structure and use of instrument drivers.

User Interface Programming

Explore additional user interface features in LabWindows/CVI and use the features in hands-on exercises. Topics include changing the appearance and function of user interface objects, creating custom menus with the menu editor, and programmatically controlling complex menus.

Interoperability and Network Communication

Learn technologies for implementing interoperability and network communication, including .NET, ActiveX, network variables, TCP, and UDP. Topics include interacting with .NET assemblies using LabWindows/CVI and ActiveX technologies, using LabWindows/CVI as an automation client to control external applications, and building applications that can pass data over the network using network variables.

Creating and Using DLLs

Learn techniques to create and use DLLs, including developing code that is modular and reusable, compiling your modules as a DLL so other applications can take advantage of your development, and easily debugging your DLLs.

Multithreading and Interface to Win32 API

Study advanced programming techniques in LabWindows/CVI. Learn how to create a multithreaded application and use thread safe queues to safely pass data between threads. Also learn to use the Interface to Win32 API functions to add flexibility to your LabWindows/CVI applications.

LabWindows/CVI Toolkits and Modules

Explore powerful features to further decrease your development time and costs, including using the LabWindows/CVI Execution Profiler Toolkit to identify bottlenecks and inefficient code and optimize run-time performance.

Suggested Next Courses

- Developing Test Programs Using TestStand
- Architecting Test Systems Using TestStand

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- **Develop Automated Production Test Systems**
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- **Semiconductor Test System Curriculum**
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Managing Software Engineering in LabVIEW

Managing Software Engineering in LabVIEW builds on the software engineering process that was discussed in the LabVIEW Core 3 course. This course expands on this knowledge by focusing on the LabVIEW tools and industry practices that aid in the definition, management, design, development, validation, and deployment of a quality solution. The course teaches common practices for managing large, team-oriented, application development projects from specification to deployment.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Manage LabVIEW projects that require the timely delivery of a quality solution that satisfies application requirements and regulatory standards
- Manage the development of a LabVIEW project from definition to deployment
- Adapt the software engineering process, as presented in this course, to your LabVIEW development projects

Prerequisites

- LabVIEW Core 3 or equivalent experience

NI Products Used

- LabVIEW Professional Development System for Windows
- Requirements Gateway
- LabVIEW VI Analyzer Toolkit
- LabVIEW Unit Test Framework Toolkit
- LabVIEW Application Builder

Coursework Topics

Introduction

Get an introduction to the software development process that defines the role of a LabVIEW Architect. Also learn about tools to monitor and control the evolution of a software product.

Requirements Gathering

Explore how to plan a LabVIEW project and learn how to identify appropriate tools for managing a project and tracking its progress. Then learn how to estimate the time required to complete a project. Lastly, create a project plan to develop a LabVIEW project.

Design

Examine tools and practices you can use to aid in designing a LabVIEW project. Learn how to develop a graphical model of the application, prototype for the user interface, and design document to use throughout the project development.

Development

Discover how to use tools in LabVIEW to improve the development of LabVIEW projects. Learn more about existing tools for automating development tasks as well as how to develop your own. Also learn how to prepare for and conduct an effective code review.

Code Reviews Validation

Review the importance of testing code for performance and functionality. Examine tools you can use to analyze and validate your code. Topics include static code analysis, dynamic code analysis, and functional code validation.

Deployment

Discover how to develop different types of build specifications depending on your deployment requirements. Also learn about the challenges that can arise when deploying your application for different types of targets and operating systems.

Suggested Next Courses

- Advanced Architectures in LabVIEW
- Object-Oriented Design and Programming in LabVIEW
- Actor-Oriented Design in LabVIEW

Validate Your Expertise With an NI Certification

Certified LabVIEW Architect (CLA)

Prerequisite: CLD
Format: Application development
Duration: 4 hours
Location: NI branch offices, online, or on-site at your location

Recertification Process

Recertification Interval: 4 years

- Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Modular Instruments 1: Digital Multimeters (DMMs)

The instructor-led, online Modular Instruments 1: Digital Multimeters (DMMs) course teaches digital multimeter fundamentals including theory and use. This course also prepares you to set up the hardware, configure the device, and program your application using LabVIEW software. In addition, the course introduces you to NI switches to help you extend the functionality of your application. The online course combines interactive learning technology through the Internet with live instructor-led lectures and hands-on exercises to deliver many of the benefits of an instructor-led classroom course while reducing the cost of training and development.

- C Classroom: 3 days
- S On-Site: half day by request
- V Virtual: one 4-hour session
- O Online: self-paced

Coursework Goals

- Configure and operate your DMM
- Understand DMM specifications (accuracy, resolution, sensitivity)
- Understand the DMM measurement cycle
- Set up signal connections to the hardware
- Use LabVIEW to program DMM applications
- Use the functions on the NI-DMM function palette
- Use the DMM to control a switch (scanning)
- Use the DMM/Switch Express VI
- Take basic measurements (voltage, current, resistance)

Prerequisites

- LabVIEW Core 1 or equivalent experience
- Basic familiarity with benchtop or handheld DMMs
- Basics circuits theory (voltage, current, resistance)

NI Products Used

- LabVIEW Professional Development System 2011 or later
- NI-DMM
- NI-SWITCH



Coursework Topics

Instrument Fundamentals

Explore the NI DMM and its different form factors and compare it with other data acquisition hardware. Topics include DMM instrumentation, DMMs versus other data acquisition hardware, and DMM benefits.

DMM Terminology

Discuss important DMM terminology, ways to calculate device accuracy, and the effects of temperature on accuracy. Also learn how to specify resolution and sensitivity.

Using the DMM

Learn how to make hardware connections to the DMM for different types of measurements and how to use the DMM Soft Front Panel. Examine the details of DMM application programming (single-point, multipoint, waveform mode).

Advanced DMM Topics

Examine the DMM measurement cycle, discuss the effects of configuration, and explore the different DMM modes.

DMMs and Switches

Discover how to use a switch to expand the capability of the DMM and how to program a DMM and switch application. Learn about the hardware connections as well as the two different scan modes used between DMMs and switches.

Suggested Next Courses

- Modular Instruments 2: Switches
- LabVIEW Core 2
- LabVIEW Core 3

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Modular Instruments 2: Switches

The instructor-led, online Modular Instruments 2: Switches course covers the fundamental concepts of switches and prepares you to set up the hardware, configure the device, and program your application using LabVIEW. The course also introduces RF switching concepts and examines NI switch modules for a wide range of applications. The online course combines interactive learning technology through the Internet with live instructor-led lectures and hands-on exercises to deliver many of the benefits of an instructor-led classroom course while reducing the cost of training and development.

- C Classroom: 3 days
- S On-Site: half day by request
- V Virtual: one 4-hour session
- O Online: self-paced

Coursework Goals

- Connect and control your NI switch device
- Understand switch specifications (relay type, topology, crosspoints, bandwidth)
- Learn important considerations for RF applications
- Use LabVIEW to program switch applications
- Use the functions on the NI-SWITCH function palette
- Understand the difference between NI-DAQmx switch functions and NI-SWITCH functions
- Use a switch with an NI DMM module
- Use the soft front panel of the device

Prerequisites

- LabVIEW Core 1 or equivalent experience
- Basic circuit theory (voltage, current, resistance)

NI Products Used

- LabVIEW Professional Development System Version 2010 or later
- Switch device
- NI-SWITCH
- NI-DAQmx
- NI-DMM



Coursework Topics

Switch Basics

Learn about the fundamental concepts of switches hardware and the types of switch relays that NI offers.

NI Switches

Nearly every test system today uses some aspect of switching. This lesson provides a basic introduction to NI's switch offering for switch selection compatible with your application.

RF Switch Concepts

Explore the important concerns when introducing a switch into an RF application. This includes a look at NI's RF switch capabilities and a discussion of how to avoid signal degradation due to attenuation and signal reflection.

Connectivity and Expansion

Many switch applications require a high number of channels or crosspoints—sometimes more than any one card offers. Discover how to connect a signal to a switch and increase your channel count through multiplexer and matrix expansion.

Control and Install NI Switches

Learn which hardware you need to control and install your PXI and SCXI switch module in the system.

Programming NI Switches

Examine the parts of the NI-SWITCH driver and how it is used to create a program in LabVIEW. Also look at programming NI switches in scanning modes.

Suggested Next Courses

- Modular Instruments 1: Digital Multimeters (DMM)
- LabVIEW Core 2
- LabVIEW Core 3

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Modular Instruments 3: High-Speed Digital I/O

The instructor-led, online Modular Instruments 3: High-Speed Digital I/O course covers the fundamental concepts of digital signals and prepares you to set up the hardware, configure the device, and program your application using LabVIEW. The online course combines interactive learning technology through the Internet with live instructor-led lectures and hands-on exercises to deliver many of the benefits of an instructor-led classroom course while reducing the cost of training and development.

- C Classroom: 3 days
- S On-Site: half day by request
- V Virtual: one 4-hour session
- O Online: self-paced

Coursework Goals

- Understand typical digital measurements
- Examine the functionality of NI digital waveform generator/analyzers
- Recognize and resolve errors in digital signals
- Identify necessary components and specifications for a digital test system
- Take measurements with an NI digital waveform generator/analyzer using the interactive toolset
- Explore the features of an NI digital waveform generator/analyzer

Prerequisites

- LabVIEW Core 1 or equivalent experience
- Basic circuit theory (voltage, current, resistance)

NI Products Used

- LabVIEW Professional Development System Version 2011 or later
- NI-HSDIO Version 1.7.5
- Digital waveform generator/analyzer device with connector and breakout board
- Digital Waveform Editor
- SignalExpress Version 2011



Coursework Topics

Digital I/O Fundamentals

Learn the fundamental concepts of digital signals and gain a deep understanding of digital waveform characteristics, terminology, and specifications. Also discover how internal and external noise can affect a digital signal and review specific techniques on how you can resolve this.

HSDIO Applications and Architecture

Explore the detailed features on all NI HSDIO devices and how to properly use them in an application. Get an overview of the SMC-based architecture of HSDIO devices and how to achieve synchronization for multiple devices using the NI-TCik technology.

Programming With NI-HSDIO

Develop a firm grasp on how to program an NI HSDIO device in LabVIEW using the NI-HSDIO driver. Explore the functions and tools to quickly set up and run an application.

HSDIO Advanced Features

Examine the unique features of NI HSDIO devices. The session includes information that is specific to certain models of NI HSDIO devices as well as the functionality of each specific model.

Suggested Next Courses

- LabVIEW Core 2
- LabVIEW Core 3
- LabVIEW Connectivity
- Data Acquisition Using NI-DAQmx and LabVIEW

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- **Develop Automated Production Test Systems**
- Develop Hardware-in-the-Loop Test Systems
- **Measure Physical Systems With Sensors or Actuators**
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Multisim Basics

The Circuit Design Suite (Multisim and Ultiboard) equips the professional PCB designer with world-class tools for schematic capture, interactive simulation, board layout, and integrated test. This course teaches the fundamentals of the Multisim integrated capture and simulation design environment. Students learn how to build a schematic and evaluate circuit performance through interactive simulation and advanced analyses while creating custom capture and simulation parts. Educators also benefit from additional customizable content specifically for electronics education. At the end of the Multisim Basics course, students can design and simulate a circuit that is ready for board layout and routing. The hands-on format of this course is the quickest way for students to become productive with Multisim.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Understand the Multisim user interface
- Use Multisim to capture circuit schematics
- Use interactive simulation to check your design
- Use virtual instruments and analyses
- Apply modular design with subcircuits, hierarchical blocks, or multipage designs
- Create custom title blocks
- Properly document your circuit designs
- Work with design variants
- Create custom components
- Co-simulate MCU projects along with SPICE
- Transfer your design to PCB layout software

Prerequisites

- Experience with Microsoft Windows
- Basic knowledge of Electronics Theory

NI Products Used

- Multisim Power Professional
- Ultiboard Power Professional

Coursework Topics

Schematic Capture

Explore the Multisim graphical user interface (GUI) and learn the Schematic Capture process to start your circuit design. Topics include setting environment preferences and placing and wiring components.

Simulation and Virtual Instruments

Review the simulation capabilities of Multisim as well as the concept-check features. Topics include types of simulation, virtual instruments, and circuit wizards.

Analyses

Learn how you can perform advanced analyses to verify your circuit design. Topics include configuring analyses and using custom expressions.

Results and Post-Processing

Get an introduction to post-processing and ways to use results from analyses to further calculate and examine data. Topics include configuring the post-processor and viewing results.

Advanced Schematic Capture

Explore the features that help you create professional-looking schematics and correctly documented circuit design. Topics include graphic annotations, comment placing, description box use, and title blocks.

Communication and Transfer

Learn the best practices for creating meaningful reports and methods for transferring your design to PCB layout software. Topics include reports in Multisim, the bill of materials, and transfer to PCB layout.

Projects and Design Sharing

Discover how to professionally architect your design. Learn to use design blocks and buses, check for electrical failures, and use the Project View. Topics include design blocks, hierarchical view, buses, and electrical rules check.

Design Variants

Examine design variants and learn how to properly configure and use variants. Also discuss use cases.

Custom Components

Explore the process for customizing or editing components and creating new components. Topics include component properties, model makers, and existing database use.

MCU Co-Simulation

Get an introduction to the Multisim MCU Module. Topics include MCU Workspace, Code Manager, Source Code Editor, Memory View, and debugging features.

Educational Features*

This optional lesson introduces the educational features of Multisim, which are mostly suitable for an academic classroom setup. Topics include rated virtual components, ladder diagrams, circuit description box, and 3D breadboarding.

* Typically for academic audiences but optional for professional audiences. Topics discussed are included in the educational edition of Multisim; however, not all of them are available in the Base, Full, or Power Pro edition.

Suggested Next Course:

- Ultiboard Basics: PCB Layout

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

“This was a great learning experience for me. I highly recommend the virtual training format.”

Charles Wilson, NRC



Object-Oriented Design and Programming in LabVIEW

Object-oriented programming has demonstrated its superiority over procedural programming for solving a variety of problems. Object-oriented design encourages cleaner interfaces between sections of code and results in code that is easier to debug and that scales better for large programming teams. Object-oriented programming is the development of code in a language that enforces object-oriented design principles. This course covers the fundamental concepts of object-oriented design and programming and then demonstrates how those concepts are implemented in LabVIEW.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: three 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Determine the appropriateness of using an object-oriented design
- Understand the object-oriented approach to developing an application
- Design an application using object-oriented design principles
- Implement a basic class hierarchy using LabVIEW classes
- Use LabVIEW features that provide additional functionality to LabVIEW classes
- Implement an application
- Examine object-oriented design patterns
- Modify an existing LabVIEW application to replace common patterns with LabVIEW objects

Prerequisites

- LabVIEW Core 3 or equivalent experience

NI Products Used

- LabVIEW Professional Development System Version 2010

Coursework Topics

Introduction

Learn how to determine if an object-oriented approach should be used for designing and implementing an application.

Designing an Object-Oriented Application

Discover how to derive a class hierarchy for an application using object-oriented design principles.

Object-Oriented Programming in LabVIEW

Explore how to develop and use classes in LabVIEW that are readable, scalable, maintainable, and reusable. Topics include encapsulation, inheritance, and dynamic dispatch.

Object-Oriented Tools and Design Patterns

Learn how to develop an object-oriented LabVIEW application that leverages one or more existing tools or design patterns. Topics include object references and construction guarantees, front panel displays for object data, and channeling and aggregation patterns.

Reviewing an Object-Oriented Application

Review, refactor, and deploy code using effective object-oriented design and programming practices.

Suggested Next Courses

- LabVIEW Core 3
- Actor-Oriented Design in LabVIEW
- Managing Software Engineering in LabVIEW
- Advanced Architectures for LabVIEW
- LabVIEW Performance
- LabVIEW Connectivity

Validate Your Expertise With an NI Certification

CLD | Certified LabVIEW Developer Certification

Prerequisite: None
 Format: Application development
 Duration: 4 hours
 Location: NI branch office, training center, online, or on-site at your location

Recertification Process
 Recertification Interval: 3 years

- Develop LabVIEW Applications**
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum**
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



RF Device Test With STS

This course follows the typical customer workflow and milestones including tight interaction with STS RF hardware. After completing this course, RF test developers will be able to use STS RF resources to interactively create, modify, execute, and debug test code modules as well as test programs based on RF configurations.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Recognize STS RF hardware components and their purposes and functions
- Create RF waveforms
- Validate device under test (DUT) RF behavior
- Define RF system calibration
- Create test programs for RF devices
- Customize RF steps

Prerequisites

- General knowledge of semiconductor test strategies and methods
- Test Program Development With STS course
- Test Code Module Development With STS course

NI Products Used

- NI STS Software Bundle (2018)
- NI Semiconductor Test System

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- **Semiconductor Test System Curriculum**
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Coursework Topics

Introduction to the RF Subsystem
Get an introduction to the RF subsystem including its components, functionality, and purpose.

Creating RF Waveforms
Explore the basics of complex RF waveforms and learn how to create them.

Validating DUT RF Behavior
Discover how to bring up the device and implement simple tests.

RF System Calibration
Learn how to define sweep settings and perform an RF calibration.

Synchronizing Instruments
Examine the triggering scheme used in the TSM RF steps.

Creating an RF Configuration-Based Test Program
Discover how to create a test program and examine its structure.

Customizing an RF Test Step
Explore how to customize an RF test step using TSM RF steps.

Exploring Instrument Resources
Learn how to use the APIs of the NI-RFmx and NI-RFPM drivers.

Suggested Next Courses

- LabVIEW Core 2
- Developing Test Programs in TestStand

Sound and Vibration Fundamentals

The instructor-led, online Sound and Vibration Fundamentals course covers key concepts and measurement techniques related to the field of sound and vibration. The online course combines interactive learning technology through the Internet with live instructor-led lectures and hands-on exercises to deliver many of the benefits of an instructor-led classroom course while reducing the cost of training and development.

- C Classroom: 3 days
- S On-Site: half day by request
- V Virtual: one 4-hour session
- O Online: self-paced

Coursework Goals

- Describe properties of linear and nonlinear systems
- Explore the uses and limitations of the FFT
- Select an appropriate window based on the application
- Describe the difference between free and forced vibration
- Evaluate a sensor based on its specifications and application
- Measure and analyze vibration data
- Contrast the different sound parameters
- Explore different sound pressure level measurements
- Implement octave analysis on sound data
- Identify different sound fields and describe how they relate to microphone selection

Prerequisites

- Basic understanding of LabVIEW and the environment; LabVIEW programming experience is not essential for this course but helpful
- Basic understanding of sound and vibration theory

NI Products Used

- LabVIEW Professional Development System Version 2009 or later
- cDAQ-9172 chassis
- NI 9236
- NI 9236 quarter-bridge input module
- LabVIEW Sound and Vibration Measurement Suite



Coursework Topics

Linear Systems and Frequency Analysis

Explore the classes of systems and linearities and nonlinearities in systems you are measuring. This information affects how you perform and interpret frequency analysis. Topics include windowing, bandwidth versus resolution, and an introduction to frequency analysis and the DFT.

Vibration and Vibration Measurements

Learn what the term vibration means and what natural frequency and resonance refer to. Explore simple harmonic motion and the spring mass model, which can be applied to the construction of an accelerometer. Also learn details about accelerometer construction, including specifications, performance, and how to match it to the performance of your measurement hardware.

Sound and Sound Measurements

Examine the concepts behind sound transmission and sound measurements. Topics include basic parameters of sound, expressing sound in terms of dB, sound pressure versus sound power, sound fields, and an introduction to octave analysis.

Suggested Next Courses

- LabVIEW Core 1 and 2
- Data Acquisition Using NI-DAQmx and LabVIEW

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Test Code Module Development With STS

This course follows the typical customer workflow and milestones including tight interaction with corresponding hardware. After completing this course, test developers will be able to use STS resources to develop and debug measurement code modules for an STS test program, create custom test steps, and perform test program optimization and deployment.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Develop custom code modules
- Debug code modules
- Understand and implement multisite subsystem execution
- Develop a test program
- Implement test program optimization
- Reduce test time
- Deploy test programs
- Create a derivative test program

Prerequisites

- General knowledge of semiconductor test strategies and methods
- Test Program Development With STS course

NI Products Used

- NI STS Software Bundle (2018)
- NI Semiconductor Test System



Coursework Topics

Developing Code Modules in LabVIEW

Create basic code modules using LabVIEW.

Exploring Tester Resources

Explore the TSM and Instrument APIs, learn how to use them in code modules, and discover how to find them in the programming environment.

Developing Test Code for an STS

Create a code module using basic APIs and the code module programming flow.

Developing Test Programs

Using STS software, control the device under test and develop a complete test program to perform all required tests.

Developing Multisite Test Programs for STS

Investigate the batch process model and modify a test program for multisite execution.

Test Time Reduction

Using the Test Execution Profiler, Test Program Performance Analyzer, and the operator interface, benchmark a test program at various levels of detail.

Deploying Test Programs

Using the TestStand Deployment Utility, deploy the complete test program and learn how to debug it after deployment.

Creating a Derivative Test Program

Make a derivative test program that avoids common pitfalls using the complete test program.

Suggested Next Courses

- RF Device Test With STS
- LabVIEW Core 2
- Developing Test Programs in TestStand

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Test Program Development With STS

This course delivers hands-on training for setting up and using an NI Semiconductor Test System (STS) to communicate with a device under test (DUT). It follows the typical semiconductor test workflow and milestones including tight interaction with corresponding hardware. After completing this course, test engineers will be able to use STS tester resources to interactively create, modify, execute, and debug test programs with prewritten code modules; collect test data; and generate test time reports.

- C** Classroom: 3 days
- S** On-Site: 3 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Set up and configure an STS to test a mixed-signal semiconductor device
- Use STS tester resources to interactively create, modify, execute, and debug a test program with prewritten code modules
- Successfully communicate with a DUT
- Understand test program architecture, modify it, and configure the execution flow
- Create an automated sequence using the Create STS Project Tool
- Modify and run a test program and gather the data
- Comfortably navigate the operator interface
- Debug devices, signals, and test sequences with debug panels
- Collect test data and generate test reports
- Benchmark test time

Prerequisites

- General knowledge of semiconductor test strategies and methods
- General computer proficiency
- Basic test engineering knowledge

NI Products Used

- NI STS Software Bundle (2018)
- NI Semiconductor Test System (STS) T1

Coursework Topics

System Overview

Get an introduction to the NI Semiconductor Test System (STS).

Exploring NI STS Instrumentation

Get an introduction to STS instrumentation and resources including their specifications, functionalities, and purposes.

Create STS Project

Explore how to use the Create STS Project Tool.

Mapping STS Hardware to DUT Pins

Discover how to create and modify a pin map.

Exploring the Digital Pattern Instrument

Learn how to perform DUT control and digital tests using the Digital Pattern Editor.

Validating DUT Behavior

Examine how to bring up the device, interactively control tester resources, and implement simple tests.

Creating and Bursting Digital Patterns

Learn how to create, load, and burst basic digital patterns.

Exploring the STS Software Development Environment

Explore how to add steps to the test sequence or call prewritten code modules.

Configuring Test Program and Steps

Examine test step templates and methods to modify test limits, configure binning, execute test programs, and report the results.

Debugging

Discover how to debug devices, signals, and the test sequence.

Using the STS Operator Interface

Learn how to run a test program in the operator interface and acquire socket time.

Suggested Next Courses

- Test Code Module Development With STS
- RF Device Test With STS

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum**
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Transitioning to LabVIEW NXG

If you are an experienced LabVIEW user who wants to develop code with LabVIEW NXG, this course helps you migrate your development skills and existing applications from LabVIEW 2017 to LabVIEW NXG. Learn the fundamental differences between the two development environments, techniques for new application development using LabVIEW NXG, and ways to use the Code Conversion Utility to migrate a qualified application for use in LabVIEW NXG.

C Classroom: 3 days

S On-Site: 3 days

V Virtual: four 4-hour sessions

O Online: self-paced

Coursework Goals

- Describe the fundamental differences between the LabVIEW 2017 and LabVIEW NXG development environments
- Develop and deploy a state machine application from scratch using LabVIEW NXG
- Use the Code Conversion Utility and LabVIEW NXG development techniques to migrate a qualified application for use in LabVIEW NXG

Prerequisites

- LabVIEW Core 1
- LabVIEW Core 2

NI Products Used

- LabVIEW NXG
- LabVIEW 2017
- NI-DAQmx
- Measurement & Automation Explorer (MAX)

Coursework Topics

Exploring the LabVIEW NXG Environment

Learn about software installation changes and explore the LabVIEW NXG environment.

Configuring Hardware

Connect to your hardware and verify its functionality in LabVIEW NXG.

Developing VIs

Discover how to develop a VI in LabVIEW NXG and prepare a VI for use as a subVI in LabVIEW NXG. Review changes to VI documentation.

Developing Object-Oriented Applications:

Create a simple object-oriented application that uses inheritance and dynamic dispatch.

Debugging and Troubleshooting

Use LabVIEW NXG tools to determine the root cause of compiler errors and debug incorrect execution.

Distributing Code:

Package LabVIEW NXG code for distribution and create a distribution for use by other developers and end-users.

Putting It All Together:

Create a State Machine

Create an application that uses a state machine to implement the functionality described by a flowchart.

Migrating Simple VIs

Learn how to migrate your existing code to LabVIEW NXG. Use the Code Conversion Utility to convert simple VIs.

Migrating an Application

Explore how the migration process may differ for larger, more complex VIs and LabVIEW applications.

Migrating Reuse Libraries

Call the Code Conversion Utility through command line instructions and use a batch file to automate the conversion of code that resides on disk.

Calling Existing VIs From LabVIEW NXG

Call existing VIs directly from LabVIEW NXG without migrating them.

Putting It All Together: Migrate an Application

Use the Code Conversion Utility and LabVIEW NXG to migrate an existing LabVIEW application to LabVIEW NXG.

Develop LabVIEW Applications

Design Smart Machines or Industrial Equipment

Develop Automated Production Test Systems

Develop Hardware-in-the-Loop Test Systems

Measure Physical Systems With Sensors or Actuators

Semiconductor Test System Curriculum

Manage Data With DIAdem

Acquire Images With Machine Vision

Design Wireless Communications Systems

Ultiboard Basics: PCB Layout

The Ultiboard Basics: PCB Layout course introduces you to the Ultiboard environment. This course prepares you to transfer Multisim schematic netlists to Ultiboard and design a PCB for export to production. Topics include design setup, precise part and trace placement, trace routing, and the optimization and use of autoplacement and autorouting. Students learn how to prepare final designs for manufacturing and export to industry-standard file formats. The hands-on approach of the Ultiboard Basics course steps you through PCB creation, from Multisim netlists to Gerber files, to teach the application of necessary skills to quickly become productive with Ultiboard.

- C** Classroom: 1 day
- V** Virtual: two 4-hour sessions
- S** On-Site: 1 day
- O** Online: self-paced

Coursework Goals

- Understand the features of the Ultiboard user interface
- Transfer designs from Multisim to Ultiboard
- Apply efficient part placement procedures
- Apply efficient trace placement procedures
- Create custom footprints (land patterns)
- Create copper areas and power planes
- Work with design constraints
- Forward and back annotate changes to and from Multisim
- Prepare your design for manufacturing

Prerequisites

- Experience with Microsoft Windows
- Basic knowledge of Electronics Theory
- Basic knowledge of PCB Layout Theory
- Basic knowledge of Multisim

NI Products Used

- Ultiboard Power Professional
- Multisim Power Professional



Coursework Topics

Introduction

Explore the Ultiboard graphical user interface and configuration options. Topics include the design process, the Ultiboard GUI, the environment preference setting, and selection filters.

Transfer and Board Design Setup

Learn how to transfer designs from Multisim to Ultiboard and ways to create or place a board outline for your PCB. Topics include board layer technology, board outline creation, and board wizards.

Parts and Placement

Discover how to place parts inside the board outline and how to create, edit, and manage footprints using the database.

Design Setup Before Routing

Examine working with and changing the netlist. Learn how to propagate changes to Multisim and set up design constraints. Topics include Netlist Editor, pin and gate swap, and trace settings and clearance setup.

Traces and Copper Areas

Get an introduction to all the trace-routing methods and guidelines for working with copper. Topics include manual routing, working with traces, and thermal relief.

Preparing for Manufacture

Review final steps you can take to better prepare your design for manufacturing and how to export your design to the Gerber format.

Suggested Next Course:

- Multisim Basics: Schematic Capture and Simulation

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Using LabVIEW and TestStand in Regulated Industries

As product complexity increases and development time compounds, software engineering processes grow even more important to the success of any test and automation project. In addition, regulatory requirements such as those in the medical device industry—not to mention the safety and business concerns involved in managing the risks inherent in many of this industry’s applications—demand efficient and reliable development processes to address business and regulatory risk considerations.

Taught by experts in the medical device regulatory space and NI products, this course helps learners navigate regulatory requirements and business demands when using LabVIEW for test and automation for medical devices. It delves deep into FDA regulatory requirements and delivers best practices for reducing the burden commonly felt during computer software validation and test method validation activities.

- C** Classroom: 3 days
- S** On-Site: 3 days by request
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Validate LabVIEW applications
- Make strategic design decisions to minimize the impact of validating and revalidating LabVIEW applications
- Implement process improvements and best practices to increase the rigor of validation yet decrease the pain associated with validation

Prerequisites

- Basic understanding of LabVIEW and the environment; LabVIEW programming experience is not essential for this course but helpful
- Some familiarity with FDA regulations and the rigors of working in the medical device industry or other highly regulated industry

NI Products Used

- LabVIEW Professional Development System
- LabVIEW Unit Test Framework Toolkit
- LabVIEW VI Analyzer Toolkit
- LabVIEW Desktop Execution Trace Toolkit
- LabVIEW Diff and LabVIEW Merge tools
- Requirements Gateway
- TestStand

Coursework Topics

Regulatory Landscape

Review the FDA’s history, purpose, and aim. Consider that smart, pain-free regulatory compliance starts with understanding the agency and why it promulgates the regulations you must follow.

The Quality System: 21 CFR 820 and ISO 13485

Learn about US and European quality system regulations and which parts apply to you as a test engineer.

21 CFR 820.70(i) and Software Validation

Explore how software that automates the quality system or the manufacturing process is validated to an intended use.

Change Management and Revalidation

Realizing change will happen, explore how revalidation does not have to be overly burdensome.

21 CFR 820.72 and Test Method Validation

Discover how test method validation can be encompassed by the software validation activity or in some cases avoided by clever design.

21 CFR 11 and Good Documentation Practices

This lesson demystifies Part 11 and gives tactical advice on how to meet Part 11 head on.

Automate DQ With VI Analyzer, LabVIEW Desktop Execution Trace Toolkit, LabVIEW Diff, and LabVIEW Merge

Learn how to streamline Design Qualification activities and take credit for good software development processes.

Automate IQ With LabVIEW Object-Oriented Programming and a HAL

Examine how to minimize tester downtime with easy to automate Installation Qualification protocols.

Automate OQ With LabVIEW Unit Test Framework

Validating that your software is meeting operational requirements is just as important and can take twice the time. Learn how to streamline operational qualification and validation with the LabVIEW Unit Test Framework.

Good Tester Design Minimizes PQ

Performance Qualification can be minimized by selecting the correct instrumentation and using it as intended. Topics include GR versus GR&R, PQ, and an instrument suitability assessment.

Trace Matrix With Requirements Gateway

Do the work up front and generate your trace matrix with the push of a button. Topics include linking requirements to tests, dragging and dropping trace relationships, and visualizing requirements coverage.

What Is TestStand?

Delve into TestStand and its key concepts from the TestStand 1 and TestStand 2 courses. These concepts are reexamined under a regulatory compliance lens.

Suggested Next Courses

- LabVIEW Core 3
- Object-Oriented Design and Programming in LabVIEW
- Managing Software Engineering in LabVIEW

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems**
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems

Using NI InsightCM™ for Condition Monitoring

The Using NI InsightCM for Condition Monitoring course delivers hands-on training for setting up and using an online condition monitoring solution that includes InsightCM and NI condition monitoring system (CMS) devices, which acquire data from sensors attached to the equipment you want to monitor. Use InsightCM Data Explorer for in-depth visualization and analysis of real-time and historical offline data and learn how to configure NI CMS devices, manage alarms, and visualize and analyze data. At the end of the course, you will be able to configure NI CMS devices, manage alarms, and visualize and analyze data.

- C** Classroom: 2 days
- S** On-Site: 2 days
- V** Virtual: four 4-hour sessions
- O** Online: self-paced

Coursework Goals

- Set up and configure NI CMS devices to acquire data from your equipment to implement a reliable condition monitoring solution
- Configure and manage alarms
- Visualize and analyze periodic, alarm, runup, and coastdown data to help determine the health of your equipment

Prerequisites

- Category II or higher vibration analyst knowledge is recommended

NI Products Used

- InsightCM
- CMS devices

- Develop LabVIEW Applications
- Design Smart Machines or Industrial Equipment
- Develop Automated Production Test Systems
- Develop Hardware-in-the-Loop Test Systems
- Measure Physical Systems With Sensors or Actuators
- Semiconductor Test System Curriculum
- Manage Data With DIAdem
- Acquire Images With Machine Vision
- Design Wireless Communications Systems



Coursework Topics

Introduction to Condition Monitoring and InsightCM:

Explore condition monitoring and the components of the InsightCM solution.

InsightCM Basics:

Set up and configure Condition Monitoring System (CMS) devices to acquire data from the equipment you want to monitor.

Vibration Analysis:

Learn about vibration analysis 101, vibration displays, and run-up/coast-down topics.

Motor Current Signature Analysis:

Learn how to use a Motor Current Signature Analysis (MCSA) system.

Thermal Imaging:

Learn how to use a thermal imaging system.

Electromagnetic Signature Analysis:

Learn how to use an EMSA system.

Advanced Server Topics:

Learn about the options for InsightCM server management including configuring roles and permissions, e-mail alerts, import/export configurations, asset templates, and more.



NI Global Services and Support

NI complements its industry-leading products with services from experts around the globe to help you achieve your goals. Whether your challenges are simple or complex, NI can help you maximize productivity and reduce costs with services such as Training, Technical Support, Consulting and Integration, and Hardware Services.



Software Services

Maximize the use of NI software by leveraging services such as technical support, online training, flexible licensing, and asset management.

Hardware Services

Minimize downtime, save on repair costs, and ensure measurement accuracy with world-class service programs for hardware.

Customer Education

Develop 50 percent faster and spend 43 percent less time on code maintenance with NI training courses. Also validate your expertise with NI certifications.

Technical Support Services

Get started with NI products faster or troubleshoot tough issues by contacting NI applications engineers who are ready to help via phone and email.

Professional Services

Leverage our extensive network of Alliance Partners and NI engineers for assistance with prototyping, feasibility analysis, consulting, and systems integration.

Technical Resources

Access volumes of self-help information at ni.com including application tips, example programs, and developer communities.

Enabling Success Every Step of the Way

Take advantage of services that meet your needs in any phase of the application life cycle, from planning and development through deployment and ongoing maintenance.

- Professional technical support
- Online and classroom-based training courses
- Software upgrades and updates
- Hardware repair and calibration
- System assembly and test
- Prototyping and feasibility analysis
- Consulting and development assistance

Proficiency Events

Learning doesn't stop after you leave the classroom. Online discussion forums, community pages, local user groups, mentorship opportunities, and proficiency events offer an invaluable resource for bolstering the essential competencies learned through training while providing a communal environment for you and your peers to engage in collaborative problem solving, technical discussions, social networking, and much more.

Certification Summits

Certification summits are complimentary events for NI-certified professionals that bring together some of the world's best LabVIEW programmers to discuss their work, share programming best practices, and network with other professionals and NI engineers. The summits' technical presentations are developed and delivered by local community members and heavily geared toward driving discussion within the group. Maintain your active certification to receive an invitation to the summits, often cited as some of the most important events of the year.

Learn more at ni.com/certification-summit.

LabVIEW Developer Days

LabVIEW Developer Days expands your LabVIEW software knowledge through NI-created technical presentations, product demonstrations, and collaborative sessions with software experts and fellow attendees. Gain confidence in your LabVIEW development skills, knowledge, and productivity through practical, in-depth presentations on principles and methodologies. Take advantage of this live event to network with peers, dive deeply into your questions, and meet local NI contacts. No certification is required.

Learn more at ni.com/labview-developer-days.

"The CLA Summit is like NIWeek for the biggest NI nerds. No basics, no introduction—just slamming up against the edge. You get a great view of what other advanced folks are doing and running into with their programming. Best use of my training hours every year."

— Matt Holt, ELM Energy, Four-Time CLA Summit Attendee

