



## One Week Workshop on LabVIEW

8<sup>th</sup> – 12<sup>th</sup> Dec 2014

The Department of Electrical & Electronics Engineering in association with National Instruments, is organizing a Workshop on LabVIEW Core 1 and Core 2, from the 8<sup>th</sup> to the 12<sup>th</sup> of December 2014.

Resource Person for the Workshop: Mr. Jason Strydom (NI South Africa)

Venue: Communications Lab, 6<sup>th</sup> Floor, Engineering Tower

Workshop Organizer: Dr Sayed Hassen, Electrical & Electronic Dept, FoE

Contact details: Email: [z.sayedhassen@uom.ac.mu](mailto:z.sayedhassen@uom.ac.mu), Tel: 403 7833

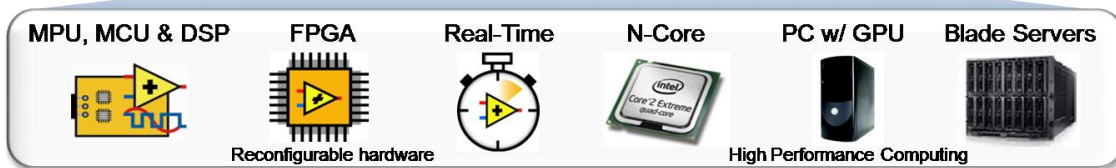
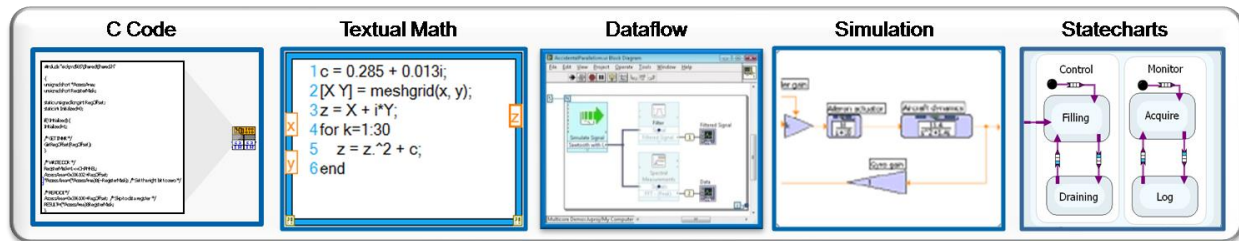
LabVIEW is a graphical programming platform that helps engineers scale from design to test and from small to large systems. It offers unprecedented integration with existing legacy software, IP, and hardware while capitalizing on the latest computing technologies. LabVIEW provides tools to solve today's problems—and the capacity for future innovation—faster and more effectively.

The LabVIEW Core 1 course introduces the LabVIEW environment, its features, dataflow programming, and common LabVIEW architectures. This course is the fastest way to become productive with LabVIEW. It prepares you to develop test and measurement, data acquisition, instrument control, data logging, and measurement analysis applications using LabVIEW.

The LabVIEW Core 2 course is an extension of the LabVIEW Core 1 course and teaches you to use common design patterns to successfully implement and develop LabVIEW applications for research, engineering, and testing environments. Topics covered include use of event-driven programming, programmatic control of your user interface, techniques to optimize reuse of existing code, and 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.



NI LabVIEW is a graphical programming environment used on campuses all over the world to deliver hands-on learning to the classroom, enhance research applications, and foster the next generation of innovators. With the intuitive nature of graphical system design, educators and researchers can design, prototype, and deploy their applications.



## Top Reasons to Use LabVIEW for Academic Teaching

### Visualize and Interact With Live Data While Your Application Is Running

Quickly create GUIs to interact with your applications and visualize your data using hundreds of included charts, graphs, thermometers, and 2D and 3D visualization tools.

### Access Hundreds of Free LabVIEW Courseware Downloads

Gain access to free educational resources, labs, exercises, and tutorials that can help you learn how to use LabVIEW and incorporate it into a multitude of engineering and science disciplines.

### Prepare Students With Industry-Standard Tools and Skills

LabVIEW is a development environment that has been built specifically for engineers and scientists and is used by over 90 percent of Fortune 500 manufacturing companies. Give your students the tools they need to be successful after graduation.

# LabVIEW Core 1

## Overview

The first step in any NI LabVIEW learning path, LabVIEW Core 1 gives you the chance to explore the LabVIEW environment, dataflow programming, and common LabVIEW development techniques in a hands-on format. Learn to develop data acquisition, instrument control, data-logging, and measurement analysis applications. At the end of the course, you will be able to create applications using the state machine design pattern to acquire, process, display, and store real-world data.

## Duration

Instructor-led Classroom: Three (3) Days

Instructor-led Online: Four (4) 4-hour sessions, plus homework

## Audience

- New users and users preparing to develop applications using LabVIEW or NI Developer Suite
- Users and technical managers evaluating LabVIEW or NI Developer Suite in purchasing decisions
- Users pursuing the Certified LabVIEW Associate Developer certification

## Prerequisites

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

## NI Products Used During the Course

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

## After attending this course, you will be able to:

- Understand front panels, block diagrams, icons, and connector panes
- Create user interfaces with charts, graphs and buttons
- Use the 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 (DAQ) devices

## Registration

Register online at [ni.com/training](http://ni.com/training) or call (800) 433-3488 Fax: (512) 683-9300 email [info@ni.com](mailto:info@ni.com)

Outside North America, contact your local NI Office.  
Worldwide Contact Info: [ni.com/global](http://ni.com/global)

## Part Number

- 910795-xx
  - 01 NI Corporate or Branch
  - 11 Regional
  - 21 Onsite (at your facility)
  - 69 Instructor-led Online
  - 71 Self-paced Online, 1 year access
  - 76 Self-paced Online, 6 month access

- Create applications that use GPIB and serial port instruments
- Use the state machine design pattern in your applications
- Use local variables to modify front panel controls or stop parallel loops

## Suggested Next Courses

- LabVIEW Core 2
- Data Acquisition and Signal Conditioning
- LabVIEW Instrument Control
- LabVIEW Real-Time 1
- LabVIEW FPGA
- Other hardware courses

# LabVIEW Core 1 Course Outline

## Day 1

### Navigating LabVIEW

This lesson introduces the LabVIEW environment. In this lesson, you will build a simple LabVIEW application that enables you to fully explore the environment and that acquires, analyzes and presents data. Topics include:

- The LabVIEW environment including windows, menus, and tools
- The LabVIEW front panel and block diagram
- Creating and using LabVIEW projects
- Understanding the dataflow programming model of LabVIEW
- Searching for controls, VIs, and functions

### Troubleshooting and Debugging VIs

This lesson teaches various debugging and error-checking techniques in LabVIEW to identify problems with block diagram organization or with data passing between different points in the block diagram.

Topics include:

- Correcting broken VIs
- Using common debugging techniques
- Addressing undefined or unexpected data
- Implementing error checking and error handling

### Implementing a VI

This lesson teaches how to create and document a VI that incorporates iterative and decision-making structures. Topics include:

- Designing a user interface (LabVIEW front panel)
- Choosing data types
- Using structures like the While Loop and For Loop
- Adding software timing to your code
- Making decisions in your code using case structures
- Documenting your code

## Day 2

### Developing Modular Applications

This lesson introduces modular programming in LabVIEW. In LabVIEW, when a VI is used within another VI, it is called a subVI. You will 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
- Creating an icon and connector pane
- Using a VI as a subVI
- Creating subVIs from an existing VI

### Creating and Leveraging Structures

This lesson introduces 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
- Creating and using cluster controls and indicators
- Using type definitions to improve reuse of data structures in applications

## Day 3

### Managing File and Hardware Resources

This lesson teaches how to use the open-modify-close model to access and control hardware and file resources in LabVIEW. Topics include:

- High-level and low-level file I/O functions available in LabVIEW
- Implementing File I/O functions to read and write data to files
- Programming with the DAQmx API
- Instrument control and programming with instrument drivers

# LabVIEW Core 1 Course Outline

## Using Sequential and State Machine Algorithms

This lesson introduces common LabVIEW design techniques and the state machine design pattern.

Topics include:

- Sequential programming
- State programming
- State machine design pattern

## Solving Dataflow Challenges Using Variables

This lesson explains how to use local variables to modify front panel control values, stop parallel loops, and circumvent dataflow limitations. Topics include:

- Communicating between parallel loops
- Using local variables
- Writing to controls and reading from indicators
- Understanding and avoiding race conditions

# LabVIEW Core 2

## Overview

The LabVIEW Core 2 course is an extension of the LabVIEW Core 1 course and teaches you to use common design patterns to successfully implement and distribute LabVIEW applications for research, engineering, and testing environments. Topics covered include use of event-driven programming, programmatic control of your user interface, techniques to optimize reuse of existing code, and 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.

## Duration

Instructor-led Classroom: Two (2) Days

Instructor-led Online: Three (3) 4-hour sessions, plus homework

## Audience

- New LabVIEW or NI Developer Suite users and users preparing to develop applications
- LabVIEW Core 1 course attendees
- Users and technical managers evaluating LabVIEW or NI Developer Suite in purchasing decisions
- Users pursuing the Certified LabVIEW Associate Developer certification

## Prerequisites

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

## NI Products Used During the Course

- LabVIEW Professional Development System
- NI Data Acquisition device
- BNC-2120

## After attending this course, you will be able to:

- Apply common design patterns that use queues and events
- Use event programming effectively
- 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

## Registration

Register online at [ni.com/training](http://ni.com/training) or call (800) 433-3488 Fax: (512) 683-9300 email [info@ni.com](mailto:info@ni.com)

Outside North America, contact your local NI Office.  
Worldwide Contact Info: [ni.com/global](http://ni.com/global)

## Part Number

910796-xx  
-01 NI Corporate or Branch  
-11 Regional  
-21 Onsite (at your facility)  
-69 Instructor-led Online  
-71 Self-paced Online, 1 year access  
-76 Self-paced Online, 6 month access

## Suggested Next Courses

- LabVIEW Core 3
- LabVIEW Connectivity
- LabVIEW Performance
- Data Acquisition and Signal Conditioning
- LabVIEW FPGA
- LabVIEW Real-Time 1
- Other hardware courses

## Suggested Certification

- Certified LabVIEW Associate Developer Certification

# LabVIEW Core 2 Course Outline

## Day 1

### Moving Beyond Dataflow with Events and Queues

This lesson describes asynchronous communication techniques for creating code that is UI-driven code and that synchronizes data between parallel loops. Topics include:

- Asynchronous communication
- Queues
- Events driven programming

### Implementing Design Patterns

This lesson introduces you to design patterns. You learn about the specific benefits and functionality of these design patterns and how they can be used to reduce development time. You learn two different categories of programming design patterns: single loop and multiple loops. Topics include:

- Design patterns
- Single loop design patterns – Including the state machine design patterns and functional global variables
- Multiple loop design patterns – Including producer/consumer design patterns
- Error handlers
- Generating error codes and messages
- Timing a design pattern

### Controlling the User Interface

This lesson describes methods to control the attributes of front panel objects programmatically, such as temporarily disabling a control. You learn how to use VI Server to access the properties and methods of front panel objects. Topics include:

- VI Server architecture
- Property nodes
- Invoke nodes
- Control references

## Day 2

### File I/O Techniques

This lesson describes different file formats for collecting and storing data and how to select the appropriate file format for your applications. You practice implementing modular code that reads or writes measurement data. Topics include:

- Comparing file formats
- Creating file and folder paths
- Writing and reading binary files
- Working with multichannel text files with headers
- Accessing Technical data management streaming (TDMS) files in LabVIEW and Excel

### Improving an Existing VI

This lesson focuses on methods to refactor inherited LabVIEW code and how to maximize reuse of existing code. Refactoring is the process of redesigning software to make it more readable and maintainable without altering its observable behavior. Topics include:

- Refactoring inherited code
- Typical issues when refactoring code

### Creating and Distributing Applications

This lesson describes the process of creating stand-alone executables and installers for LabVIEW applications. You will learn how to use the Application Builder in LabVIEW. Topics include:

- Preparing the files
- Creating build specifications
- Creating and debugging an application
- Creating an Installer