Object-Oriented Design and Programming in LabVIEW™ Exercises

Course Software Version 2010
November 2010 Edition
Part Number 325617A-01

Worldwide Technical Support and Product Information
ni.com

National Instruments Corporate Headquarters
11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices
Australia 1800 300 800, Austria 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 5050 9800, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511, France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466, New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 328 90 10, Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the Info Code feedback.

Contents

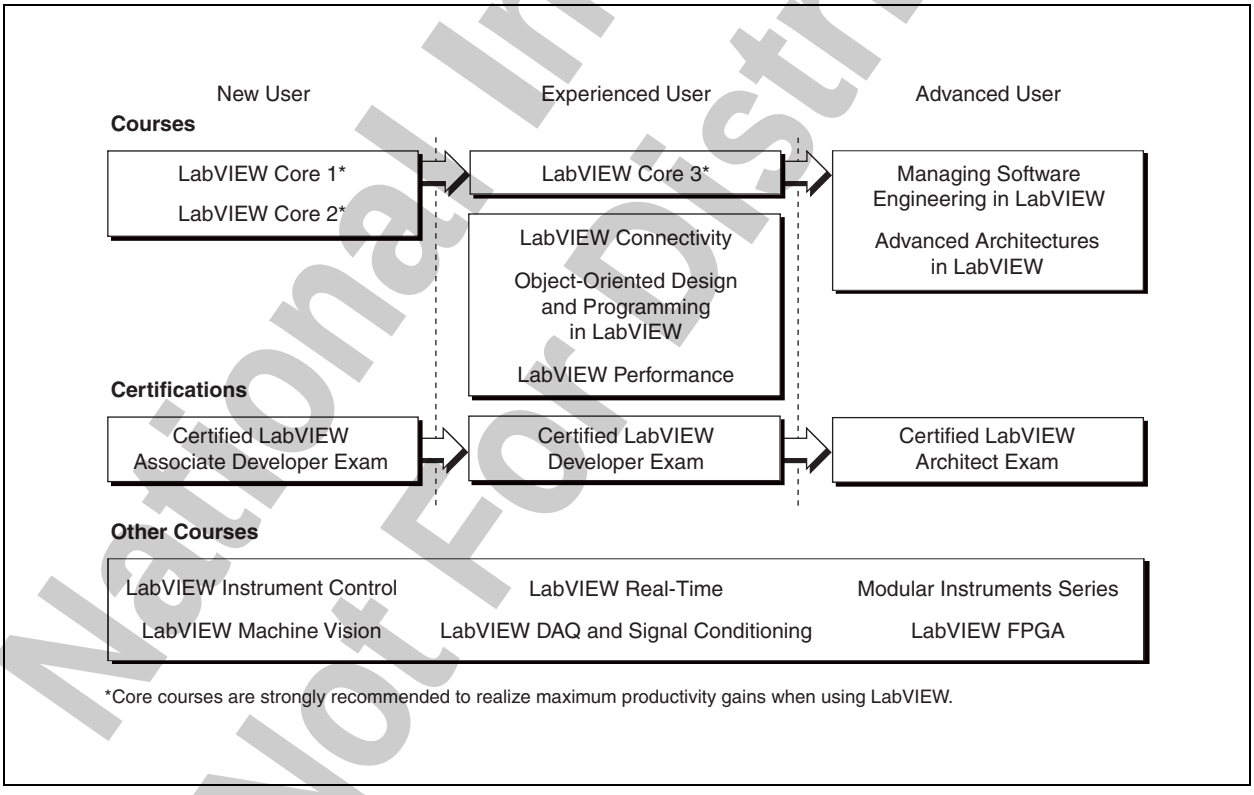
Student Guide

Thank you for purchasing the *Object-Oriented Design and Programming in LabVIEW* course kit. This course manual and the accompanying software are used in the two-day, hands-on *Object-Oriented Design and Programming with LabVIEW* course.

You can apply the full purchase price of this course kit toward the corresponding course registration fee if you register within 90 days of purchasing the kit. Visit `ni.com/training` to register for a course and to access course schedules, syllabi, and training center location information.

A. NI Certification

The Object-Oriented Design and Programming in LabVIEW course is part of a series of courses designed to build your proficiency with LabVIEW and help you prepare for exams to become an NI Certified LabVIEW Developer and NI Certified LabVIEW Architect. The following illustration shows the courses that are part of the LabVIEW training series. Refer to `ni.com/training` for more information about NI Certification.



B. Course Description

This course assumes that you have taken the *LabVIEW Core 3* course or have equivalent experience.

The course is divided into lessons, each covering a topic or a set of topics. Each lesson consists of the following parts:

- An introduction that describes what you will learn
- A discussion of the topics
- A set of exercises that reinforces the topics presented in the discussion

  Some lessons include optional exercises or challenge steps to complete if time permits
- A summary that outlines important concepts and skills taught in the lesson

Note For course manual updates and corrections, refer to `ni.com/info` and enter the Info Code `lvoop`.

C. What You Need to Get Started

Before you use this manual, make sure you have the following items:

- ❑ Windows XP or later
- ❑ LabVIEW Professional Development System 2010 or later
- ❑ *Object-Oriented Design and Programming in LabVIEW* CD, which contains the following folders and files:

Folder Name	Description
`Exercises`	Contains all the VIs and support files needed to complete the exercises in this course
`Solutions`	Contains completed versions of the VIs you build in the exercises for this course

D. Installing the Course Software

Complete the following steps to install the course software.

1. Insert the course CD in your computer. The **Object-Oriented Design and Programming in LabVIEW Course Setup** dialog box appears.
2. Click **Install the Course Material**.
3. Follow the onscreen instructions to complete installation and setup.

Exercise files are located in the `<Exercises>\Object-Oriented Design and Programming in LabVIEW` folder.

Tip Folder names in angle brackets, such as `<Exercises>`, refer to folders on the root directory of your computer.

Repairing or Removing Course Material

You can repair or remove the course material using the Add or Remove Programs feature on the Windows Control Panel. Repair the course manual to overwrite existing course material with the original, unedited versions of the files. Remove the course material if you no longer need the files on your computer.

E. Course Goals

This course prepares you to:

- Determine the appropriateness of using an object-oriented approach to solve the problem
- Design an application using object-oriented design principles
- Implement a basic class hierarchy using LabVIEW classes
- Modify an existing application written in G to replace common patterns with objects

This course does *not* present the following topic:

- Developing a complete application for any student in the class; refer to the NI Example Finder, available by selecting **Help»Find Examples**, for example VIs you can use and incorporate into VIs you create

F. Course Conventions

The following conventions are used in this course manual:

»	The **»** symbol leads you through nested menu items and dialog box options to a final action. The sequence **Options»Settings»General** directs you to pull down the **Options** menu, select the **Settings** item, and select **General** from the last dialog box.
	This icon denotes a tip, which alerts you to advisory information.
	This icon denotes a note, which alerts you to important information.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross-reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you enter from the keyboard, sections of code, programming examples, and syntax examples. This font also is used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.
`<Exercises>/.../`	This short-hand denotes the `Object-Oriented Design and Programming in LabVIEW` directory located in the `Exercises` directory.
`<Solutions>/.../`	This short-hand denotes the `Object-Oriented Design and Programming in LabVIEW` directory located in the `Solutions` directory.

2

Designing an Object-Oriented Application

Exercise 2-1 Identify Classes and Methods

Goal

Identify the classes required to complete the course project.

Scenario

You have been asked to develop a demo application for a company that makes devices for playing sound. You meet with a company representative who explains the job. The company makes two devices, a sound card that actually plays sound and a visualizer that renders the content of the sound wave visually. You notice the representative refers to both devices as “sound players.”

They want you to create a demo program that a sales employee could use to show off the company’s players. The application should allow the user to choose which of the two players to demonstrate, provide the ability for the user to configure the chosen player, and finally run a demonstration on that player.

While offering you the job, the company representative mentions that the company has other sound players in development, and if you do well on this project, you may be asked to develop demos for future products. Because you are confident you will get those future jobs, you decide that you need to make the first demo flexible so you can generate demos for those future products quickly and with as much code reuse as possible.

The representative lists several sounds that the demo should be able to perform on the players. Unfortunately, the exact order of these sounds is something that will be decided late in the development cycle due to disagreements within the company about what sounds best demonstrate the capabilities of the players. You think you might be asked to make a variety of late-in-development changes to the sounds themselves, so it seems wise to develop a framework for quickly plugging in new sounds, instead of just hardcoding a number of waveforms.

The company provides you with the programming API for both players. When you review them you can tell that although they are similar, these

APIs were written by different programming teams that did not coordinate with each other. Writing one application that can run either player will not be as simple as putting a case structure around subVI calls.

When you get back to your office, you take your notes from the customer meeting and rewrite them to be very explicit about what it is you think you are being asked to build, rewording the customer's sometimes vague requirements into more specific technical requirements.

Use this document to identify the classes that will be needed to complete the project and the methods that will be needed for each class.

Design

Identify the nouns in the requirements document. You will use these to determine the classes required to complete the project.

Identify the action verbs from the requirements document. You will use these to define the methods for each class you identified.

Implementation

Part A: Identify Potential Classes

Identify all of the potential classes for the Sound Player Demo project by reviewing the problem statement and identifying all of the nouns.

1. Review the Sound Player Demo Problem Statement.

Sound Player Demo Project Problem Statement

Develop an application that demonstrates a sound playing device. When the application runs, it should:

1. Allow the user to select the sound player to be demonstrated.
2. Allow the user to configure that player (different players require different configuration options).
3. Play the fixed set of chosen sounds on the player.
4. Shut down the system when the demo has completed.

Your demo application will show off one of two types of sound player: a sound card or a visualizer. At the start of the application, the demo will offer the user the choice of which player to demonstrate.

The application must direct the user to setup the application's sound player. Per the sound card API, the user must choose a device ID, the number of samples per channel, the sound format and the volume. The sound card will