Advanced Architectures in LabVIEW

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

Duration
Classroom: Three (3) Days

Audience
- LabVIEW programmers interested in learning advanced design patterns
- LabVIEW programmers managing large application development
- LabVIEW Architects managing a team of developers
- LabVIEW users pursuing the Certified LabVIEW Architect certification

Prerequisites
- LabVIEW Core 3 or equivalent LabVIEW experience

NI Products Used During the Course
- LabVIEW Professional Development System Version 2011 or later

After attending this course, you will be able to:
- 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 inter-process communication method

- Design a consistent, organized, and usable API
- Analyze, critique, and improve the architecture of a LabVIEW application

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

Suggested Next Certification
- Certified LabVIEW Architect Exam

Advanced Architectures in LabVIEW

Architecting an Application

In this lesson, you learn how to design and document scalable, readable, and maintainable software architectures. Topics include:

- Characteristics of scalable, readable, maintainable architecture
- Documenting a software architecture
- Characteristics of a scalable design pattern

Designing an API

In this lesson, you learn how to design a consistent, organized, and usable API that may 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
- Passing data in an API

Multiple Processes and Inter-Process Communication

In this crucial lesson, you learn about foundational APIs and design patterns and how they apply to several essential advanced design patterns. You also learn several advanced methods for communicating between multiple processes and their tradeoffs. Exercises and tools are general enough that you can use them in your own applications. Topics include:

- Storing data, streaming data, and sending messages
- Foundational native LabVIEW APIs for messaging (queues, notifiers, user events, data value references)
- Foundational design patterns (FGV, various state machines, producer/consumer, various queue-driven message handlers)
- Scaling foundational design patterns for multiple processes (e.g. client/server)
- Asynchronous dynamic processes

- By reference inter-process data storage (single element queues, data value references)
- Several inter-process communication methods
- Exposure to relevant native LabVIEW APIs, advanced design patterns, and reference architectures

Advanced User Interface Techniques

In this lesson, you learn how to create an architecture that provides a modular, scalable, and extensible user interface. Topics include:

- Subpanels
- XControls

Introduction to Object-Oriented Programming in LabVIEW

In this lesson, you learn how using the encapsulation and inheritance capabilities of LabVIEW Object-Oriented Programming can create very scalable and extensible software architectures. Topics include:

- Introduction to LabVIEW classes and related terminology
- Using encapsulation to restrict access to class data
- Using inheritance and dynamic dispatch to implement scalable polymorphism at run time

Plug-In Architectures

In this lesson, you learn how to create a plug-in architecture that allows you to add features without changing your main code. Topics include:

- Plug-in architecture using VI Server
- Plug-in architecture using LVOOP

Tips, Tricks, and Other Techniques

In this lesson, you learn various tips, tricks, and techniques to help create your architecture. Topics include:

- Variant attributes
- Callback VIs with user events
- VI Scripting
- Drop In VIs

Advanced Architectures in LabVIEW

Error Handling

In this lesson, you learn 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
- Example LabVIEW error handling reference library

Additional Resources

In this concluding section, you learn where you to find additional code and ideas. Topics include:

- NI Community groups
- Other forums