

# LV Mastery Professional: Course Outline

These pages are intended to assist you in taking notes as you complete lessons, and to assist in organizing and tracking your progress. They are printable, so you can create as many as necessary.

Category	Topics	Content
<b>Review Exercise</b>		<ul style="list-style-type: none"> <li>Selected topics from the Beginner course.</li> </ul>
<b>Typedefs and State Machines II</b>	Creating Typedefs	<ul style="list-style-type: none"> <li>What is a typedef?</li> <li>Methods for creating typedefs.</li> <li>Strict Typedefs.</li> </ul>
	Enum-Driven State Machines	<ul style="list-style-type: none"> <li>Benefits of using a typedef Enum as the state.</li> </ul>
	Array-Based State Machines	<ul style="list-style-type: none"> <li>What is an array-based state machine?</li> <li>Advantages and disadvantages.</li> </ul>
	Multi-line String Based State Machine	<ul style="list-style-type: none"> <li>What is a multi-line string based state machine?</li> <li>Advantages and disadvantages.</li> </ul>
	State Machine with User Interaction	<ul style="list-style-type: none"> <li>Programming software to allow for user interaction</li> </ul>
	Data Storage: Functional Global	<ul style="list-style-type: none"> <li>Defining functional globals.</li> <li>Using functional globals.</li> </ul>
	Functional Global Example	
<b>Exercise 1</b>		<ul style="list-style-type: none"> <li>Functional globals and state machines.</li> </ul>
<b>Advanced Loop Architectures</b>	The "Smart Stack"	<ul style="list-style-type: none"> <li>Expanding the functionality of the functional global beyond it's basic functions (initialize, stack, and use) to include functions such as: analyze, query, and save.</li> </ul>
	Multiple Loop Architectures	<ul style="list-style-type: none"> <li>Modifying state machines with advanced loop architectures.</li> <li>Types of loop architectures.</li> </ul>
	Notifiers	<ul style="list-style-type: none"> <li>Send single commands from one loop/VI to another.</li> <li>Notifier functions.</li> </ul>
	Queues	<ul style="list-style-type: none"> <li>Send multiple commands from one loop/VI to another.</li> <li>Queue functions.</li> </ul>
<b>Exercise 2</b>		<ul style="list-style-type: none"> <li>Multiple loop architectures.</li> </ul>
<b>General Programming Techniques</b>	Documentation	<ul style="list-style-type: none"> <li>Adding descriptions to your code.</li> </ul>
	LabVIEW Project Screen and File Structure	<ul style="list-style-type: none"> <li>The LabVIEW Project Explorer.</li> <li>Things to consider when making file structures.</li> <li>Advantages and disadvantages.</li> </ul>
	Error Handling	<ul style="list-style-type: none"> <li>Purpose of error handling.</li> <li>Error-In and Error-Out clusters.</li> </ul>
	Shift Register Storage Clusters	<ul style="list-style-type: none"> <li>Using a shift register to store important data between loop iterations.</li> <li>Role of typedefs.</li> </ul>
	Additional Topics	<ul style="list-style-type: none"> <li>Value (Signaling) Property Node.</li> <li>First Call? Function.</li> <li>Variant data: converting to/from.</li> </ul>
<b>Exercise 3</b>		<ul style="list-style-type: none"> <li>General Programming Techniques.</li> </ul>
<b>Advanced DAQ</b>	Review of DAQ Assistant Express VI	<ul style="list-style-type: none"> <li>Reviewing Measurement and Automation Explorer (MAX).</li> <li>Reviewing The DAQ Assistant Express VI.</li> </ul>
	Using the DAQ Assistant	<ul style="list-style-type: none"> <li>Two methods to generate code using the DAQ assistant</li> <li>One-Time DAQ.</li> <li>Continuous run DAQ.</li> </ul>
	DAQmx Functions I	<ul style="list-style-type: none"> <li>Channels and Tasks.</li> <li>Key DAQmx functions.</li> </ul>
	DAQmx Functions II	<ul style="list-style-type: none"> <li>More important DAQmx functions.</li> </ul>
	Analog Generation	<ul style="list-style-type: none"> <li>Software timed generation.</li> <li>Hardware timed generation.</li> <li>Non-regeneration.</li> </ul>
	Digital I/O	<ul style="list-style-type: none"> <li>Channels vs. Ports.</li> <li>Timing.</li> </ul>

		<ul style="list-style-type: none"> <li>• Triggering.</li> <li>• Digital Output vs. Continuous digital output.</li> </ul>
	Counter I/O	<ul style="list-style-type: none"> <li>• Counting digital events.</li> <li>• Pulse train measurement.</li> <li>• Pulse train generation.</li> <li>• Changing PWM output while running.</li> </ul>
	Synchronization	<ul style="list-style-type: none"> <li>• Same device: multiple functions</li> <li>• Across devices</li> </ul>
	Using MAX with LabVIEW	<ul style="list-style-type: none"> <li>• Advantages and Disadvantages.</li> <li>• DAQmx storage VIs.</li> <li>• Creating scales in LabVIEW.</li> <li>• DAQmx System property node.</li> </ul>
<b>Exercise 4</b>		<ul style="list-style-type: none"> <li>• Data Acquisition.</li> </ul>
<b>Building Applications</b>	Project Window	<ul style="list-style-type: none"> <li>• Using the project explorer.</li> <li>• Create new build specifications.</li> </ul>
	Applications (EXEs)	<ul style="list-style-type: none"> <li>• Selecting source files.</li> <li>• Select EXE configuration Info.</li> <li>• Debugging.</li> </ul>
	Installer	<ul style="list-style-type: none"> <li>• Deploy EXEs with Standard installers.</li> <li>• Installer settings.</li> </ul>
	DLL	<ul style="list-style-type: none"> <li>• Use your code inside other C/VB/LabVIEW Programs.</li> </ul>
	Source Distribution and ZIP File	<ul style="list-style-type: none"> <li>• Save all project files at once.</li> <li>• Apply VI Settings.</li> <li>• Create Zip Files.</li> </ul>
<b>Exercise 5</b>		<ul style="list-style-type: none"> <li>• Building Applications.</li> </ul>