

Tasks and Objectives:
Certified LabVIEW Developer

Certification ID	Certification Title	Job Description:
CLD	Certified LabVIEW Developer	Given a specification or a measurement need, the Certified LabVIEW Developer is able to create a moderately complex solution using LabVIEW

Task ID	Task	Objective ID	Objective
CLD-VPP-01	Navigate the LabVIEW Environment		
		CLD-VPP-01-01	List and identify the three parts of a virtual instrument: <ul style="list-style-type: none"> ▪ Front Panel ▪ Block Diagram ▪ Icon/Connector Pane
		CLD-VPP-01-02	Explain the operation of front panel toolbar buttons
		CLD-VPP-01-03	Explain the operation of block diagram toolbar buttons
		CLD-VPP-01-04	Identify the three LabVIEW Palettes: <ul style="list-style-type: none"> ▪ Controls Palette ▪ Functions Palette ▪ Tools Palette

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-01-05	<p>Explain the functions associated with the three LabVIEW Palettes:</p> <ul style="list-style-type: none"> ▪ Controls Palette ▪ Functions Palette ▪ Tools Palette
		CLD-VPP-01-06	Explain the function of the Right-Click Menus
		CLD-VPP-01-07	State the methods for creating Front Panel objects
		CLD-VPP-01-08	Define Dataflow Programming
		CLD-VPP-01-09	Explain the components and techniques for wiring Block Diagram objects
		CLD-VPP-01-10	<p>Define the data types used in LabVIEW</p> <ul style="list-style-type: none"> ▪ Numeric ▪ Boolean ▪ String ▪ Waveform
CLD-VPP-02	Develop LabVIEW VI Front Panels		
		CLD-VPP-02-01	<p>Describe the operation and use of the following LabVIEW controls and indicators</p> <ul style="list-style-type: none"> ▪ Numeric ▪ Boolean ▪ String ▪ Table

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-02-02	Describe the operation and function of the following indicators <ul style="list-style-type: none"> ▪ Waveform Charts ▪ Waveform Graphs ▪ X-Y Graphs
		CLD-VPP-02-03	Explain the methods for customizing Charts and Graphs including: <ul style="list-style-type: none"> ▪ Update Mode ▪ Plot Legend ▪ Scale Legend ▪ Graph Palette ▪ Cursor Palette
		CLD-VPP-02-04	Create Single and Multiple Plot Charts and Graphs, including graphs with custom X0 and delta X values
		CLD-VPP-02-05	Describe the following options available on front Panel Controls and Indicators <ul style="list-style-type: none"> ▪ Visible Items ▪ Make Current Value Default ▪ Reinitialize to default ▪ Synchronous Display ▪ Enabled State ▪ Autoscale
		CLD-VPP-02-06	Explain the characteristics of Front Panel Controls and Indicators <ul style="list-style-type: none"> ▪ Representation ▪ Format and Precision ▪ Data Range

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-02-07	Describe the function and use of the controls and indicators available for LabVIEW Front Panels
CLD-VPP-03	Develop LabVIEW VI Block Diagrams		
		CLD-VPP-03-01	Describe the function and operation of the following LabVIEW Structures <ul style="list-style-type: none"> ▪ While Loops ▪ For Loops
		CLD-VPP-03-02	Describe the function and operation of the following LabVIEW Structures <ul style="list-style-type: none"> ▪ Case Structures ▪ Sequence Structures ▪ Formula Nodes
		CLD-VPP-03-03	Identify and explain the characteristics of <ul style="list-style-type: none"> ▪ Tunnels ▪ Initialized Shift Registers ▪ Un-initialized Shift Registers
		CLD-VPP-03-04	Describe the mechanical action options for Boolean Switches <ul style="list-style-type: none"> ▪ Switch When Pressed ▪ Switch When Released ▪ Switch Until Released ▪ Latch When Pressed ▪ Latch When Released ▪ Latch Until Released
		CLD-VPP-03-05	Identify and explain the impact of coercion dots

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-03-06	Describe the array creation process <ul style="list-style-type: none"> ▪ 1D Arrays ▪ 2D Arrays ▪ Array Constants
		CLD-VPP-03-07	Explain the operation of Auto-indexing with: <ul style="list-style-type: none"> ▪ For Loops ▪ While Loops
		CLD-VPP-03-08	Explain the function, operation, inputs and outputs of Array Functions <ul style="list-style-type: none"> ▪ Array Size ▪ Index Array ▪ Replace Array Subset ▪ Initialize Array ▪ Build Array ▪ Replace Array Subset ▪ Array to Cluster ▪ Cluster to Array
		CLD-VPP-03-09	Describe Polymorphism and it's effect on array and scalar inputs
		CLD-VPP-03-10	Create Clusters <ul style="list-style-type: none"> ▪ Controls ▪ Indicators ▪ Constants
		CLD-VPP-03-11	Describe Cluster Order, and the process for changing the order

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-03-12	<p>Explain the function, operation, inputs and outputs of Cluster Functions</p> <ul style="list-style-type: none"> ▪ Bundle ▪ Unbundle ▪ Bundle by Name ▪ Unbundle by Name
		CLD-VPP-03-13	Compare and Contrast Clusters and Arrays
		CLD-VPP-03-14	<p>List the four String Display Types</p> <ul style="list-style-type: none"> ▪ Normal ▪ \ Code ▪ Password ▪ Hex Display
		CLD-VPP-03-15	<p>Explain the function, operation, inputs and outputs of the following string functions</p> <ul style="list-style-type: none"> ▪ String Length ▪ Concatenate Strings ▪ String Subset ▪ Match Pattern ▪ Format into String ▪ Scan From String
		CLD-VPP-03-16	Generate format strings using the Edit Format String option
		CLD-VPP-03-17	<p>Describe the purpose and use of the following LabVIEW error handling components:</p> <ul style="list-style-type: none"> ▪ Error Cluster ▪ Simple Error Handler

Tasks and Objectives:
Certified LabVIEW Developer

CLD-VPP-04	Debug VIs using LabVIEW Debugging tools		
		CLD-VPP-04-01	Determine whether a VI is executable or non-executable
		CLD-VPP-04-02	Given a VI block diagram, identify the cause of error conditions
		CLD-VPP-04-03	Obtain information using LabVIEW Debugging Tools <ul style="list-style-type: none"> ▪ Single-Step ▪ Probes ▪ Breakpoints
		CLD-VPP-04-04	Explain the function and operation of Execution Highlighting
CLD-VPP-05	Locate Information in the LabVIEW Help System		
		CLD-VPP-05-01	Obtain context sensitive help for a LabVIEW object
		CLD-VPP-05-02	Perform searches of LabVIEW Help Files
		CLD-VPP-05-03	Locate information contained in the online LabVIEW User Manuals
CLD-VPP-06	Create and Use Sub VIs		
		CLD-VPP-06-01	Recall the steps required to create a SubVI
		CLD-VPP-06-02	Explain the setup and use of a connector pane

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-06-03	Explain the process for creating an Icon
		CLD-VPP-06-04	Edit a SubVI
		CLD-VPP-06-05	Create a SubVI from sections of a VI
CLD-VPP-07	Perform ASCII File I/O in LabVIEW		
		CLD-VPP-07-01	Compare and contrast the capabilities of the High Level File I/O VIs and the Intermediate Level File I/O VIs
		CLD-VPP-07-02	Describe the function, operation, inputs and outputs for the following File I/O functions <ul style="list-style-type: none"> ▪ Open/Create/Replace ▪ Write File ▪ Read File ▪ Close File
		CLD-VPP-07-03	Describe the function, operation, inputs and outputs for the following High-Level File I/O VIs <ul style="list-style-type: none"> ▪ Write to Spreadsheet File ▪ Read From Spreadsheet File
CLD-VPP-08	Select Data Acquisition Hardware		
		CLD-VPP-08-01	Define the purpose of Data Acquisition components <ul style="list-style-type: none"> ▪ Transducers

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Signal Conditioning ▪ DAQ Hardware
		CLD-VPP-08-02	<p>Describe the following characteristics of DAQ hardware and their effect on measurements</p> <ul style="list-style-type: none"> ▪ Resolution ▪ Range ▪ Gain ▪ Sampling Rate ▪ Aliasing
		CLD-VPP-08-03	<p>Define the three measurement modes</p> <ul style="list-style-type: none"> ▪ Referenced Single Ended ▪ Non-referenced single Ended ▪ Differential
		CLD-VPP-08-04	<p>Determine the proper connection mode for transducer inputs</p> <ul style="list-style-type: none"> ▪ Grounded ▪ Floating ▪ High / Low Channel Counts
CLD-VPP-09	Configure and Test DAQ Hardware and Software using the Measurement & Automation Explorer (MAX)		

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-09-01	Describe the purpose of the four functional sections of MAX <ul style="list-style-type: none"> ▪ Data Neighborhood ▪ Devices and Interfaces ▪ Scales ▪ Software
		CLD-VPP-09-02	Configure Virtual Channels using MAX
		CLD-VPP-09-03	Test Virtual Channels using MAX
		CLD-VPP-09-04	Describe the Device Properties available in MAX
		CLD-VPP-09-05	Determine DAQ Device Settings using MAX
		CLD-VPP-09-06	Test DAQ Device functionality using the MAX Test Panels
		CLD-VPP-09-07	Given a desired input and output range, create a custom scale using MAX
		CLD-VPP-09-08	Explain the function and use of the DAQ Channel Name Control
CLD-VPP-10	Perform Analog Input and Output operations using the Easy Analog I/O VIs		
		CLD-VPP-10-01	Describe the purpose and organization of the Analog Input VIs <ul style="list-style-type: none"> ▪ Easy Analog Input VIs ▪ Intermediate Analog Input VIs ▪ Analog Input Utility VIs

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Advanced Analog Input VIs
		CLD-VPP-10-02	<p>Describe the function, inputs and outputs for the following analog input VIs</p> <ul style="list-style-type: none"> ▪ AI Sample Channel VI ▪ AI Sample Channels VI ▪ AI Acquire Waveform VI ▪ AI Acquire Waveforms VI
		CLD-VPP-10-03	<p>Differentiate between hardware-timed acquisition and software-timed acquisition</p>
		CLD-VPP-10-04	<p>Create single-point analog input VIs using</p> <ul style="list-style-type: none"> ▪ AI Sample Channel VI ▪ AI Sample Channels VI
		CLD-VPP-10-05	<p>Create waveform analog input VIs using</p> <ul style="list-style-type: none"> ▪ AI Acquire Waveform VI ▪ AI Acquire Waveforms VI
		CLD-VPP-10-06	<p>Describe the function, inputs and outputs of the Following Analog Output VIs</p> <ul style="list-style-type: none"> ▪ AO Update Channel ▪ AO Generate Waveform
		CLD-VPP-10-07	<p>Create single-point analog output VIs using the AO Update Channel VI</p>
		CLD-VPP-10-08	<p>Create waveform analog output VIs using the AO Generate Waveform VI</p>

Tasks and Objectives:
Certified LabVIEW Developer

CLD-VPP-11	Perform Digital Input and Output operations using Easy Digital I/O VIs		
		CLD-VPP-11-01	Describe the function, inputs and outputs for the following Digital I/O VIs <ul style="list-style-type: none"> ▪ Write to Digital Line VI ▪ Write to Digital Port VI ▪ Read from Digital Line VI ▪ Read from Digital Port VI
		CLD-VPP-11-02	Create digital input VIs using the following functions <ul style="list-style-type: none"> ▪ Read from Digital Line VI ▪ Read from Digital Port VI
		CLD-VPP-11-03	Create digital output VIs using the following functions <ul style="list-style-type: none"> ▪ Write to Digital Line VI ▪ Write to Digital Port VI
CLD-VPP-12	Perform Counter/Timer operations in LabVIEW		
		CLD-VPP-12-01	Describe the typical operations performed with Counter Timers <ul style="list-style-type: none"> • Event Counting • Frequency Measurement • Period Measurement • Position Measurement • Pulse Generation

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-12-02	Given a set of conditions and requirements, create a VI that counts event input pulses
CLD-VPP-13	Control Instruments in LabVIEW		
		CLD-VPP-13-01	State the requirements for GPIB communications including <ul style="list-style-type: none"> ▪ Maximum cable length ▪ Primary addressing values ▪ Minimum number of powered devices
		CLD-VPP-13-02	Identify the standard cable used for GPIB communications
		CLD-VPP-13-03	Connect multiple GPIB devices for communications with a PC
		CLD-VPP-13-04	Describe the purpose of Virtual Instrumentation Software Architecture (VISA)
		CLD-VPP-13-05	Define the following terms <ul style="list-style-type: none"> ▪ VISA Resource ▪ VISA Session
		CLD-VPP-13-06	Define the following terms associated with Serial Communications <ul style="list-style-type: none"> ▪ Bits Per Second ▪ Start Bit ▪ Stop Bits ▪ Data Bits ▪ Parity Bit
		CLD-VPP-13-07	Compare and contrast GPIB and Asynchronous Serial Communications

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-13-08	Create VIs using VISA that communicate with stand-alone instruments
CLD-VPP-14	Customize VI Properties		
		CLD-VPP-14-01	Describe the options available in the following VI Properties <ul style="list-style-type: none"> ▪ Window Appearance ▪ Window Size ▪ Execution
		CLD-VPP-14-02	Using VI properties, create a VI that behaves as a pop-up dialog box
		CLD-VPP-14-03	Explain the difference between using SubVI Node Setup and VI Properties
			Describe the precautions and limitations associated with the following properties <ul style="list-style-type: none"> ▪ Run When Opened ▪ Remove Diagrams
CLD-VPP-15	Design/Plan an Application in LabVIEW		
		CLD-VPP-15-01	Describe a top-down design methodology for software development including the following aspects <ul style="list-style-type: none"> • Define Project • Design Flowchart • Implement Nodes as SubVIs • Test SubVIs

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> • Integrate SubVIs into Project • Test & Release Final Project • Process Customer Feedback
		CLD-VPP-15-02	<p>Discuss the development considerations for large software projects</p> <ul style="list-style-type: none"> • Define system requirements • Determine end user expectations • Document the purpose of the application • Plan for future modifications and additions
		CLD-VPP-15-03	Design a Flow Diagram for a LabVIEW application
		CLD-VPP-15-04	<p>Implement a LabVIEW application from a Flow Diagram</p> <ul style="list-style-type: none"> • Develop VI's that correspond to flowchart nodes • Use modularization to make development easier to control • Use bottom-up development to build and test the modules and create the complete application
CLD-VPP-16	Implement Error Handling in LabVIEW VIs		
		CLD-VPP-16-01	<p>Describe the function, inputs and outputs of the following Error VIs</p> <ul style="list-style-type: none"> ▪ General Error Handler ▪ Merge Errors VI
		CLD-VPP-16-02	Explain the purpose of error cluster routing in a VI
		CLD-VPP-16-03	Generate a VI that incorporates error handling and reporting

Tasks and Objectives:
Certified LabVIEW Developer

CLD-VPP-17	Choose an appropriate VI architecture		
		CLD-VPP-17-01	Describe the common LabVIEW programming architectures: <ul style="list-style-type: none"> • Simple VI • General VI • Parallel Loop • Multiple Case Structure • State Machine
		CLD-VPP-17-02	Explain the advantages and disadvantages of the common LabVIEW programming architectures <ul style="list-style-type: none"> • Simple VI • General VI • Parallel Loop • Multiple Case Structure • State Machine
		CLD-VPP-17-03	Given a set of conditions, choose the appropriate architecture for a LabVIEW application
CLD-VPP-18	Create and Use VI Templates		
		CLD-VPP-18-01	Explain the advantages of using templates in an application
		CLD-VPP-18-02	Create a LabVIEW VI Template
		CLD-VPP-18-03	Create a LabVIEW VI from a Template
CLD-VPP-19	Design VI Front Panels		

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-19-01	Describe user interface issues associated with the following front panel attributes <ul style="list-style-type: none"> • Color • Spacing and Alignment • Text and Fonts • Decorations
		CLD-VPP-19-02	Explain the philosophy for the proper use of Controls and indicators for logical Front Panel design <ul style="list-style-type: none"> • Dialog Controls • Tab Controls • Menus • Boolean Cluster Menus
		CLD-VPP-19-03	Given a set of requirements, design a Front Panel using good user interface techniques, including: <ul style="list-style-type: none"> • Color • Spacing and Alignment • Text and Fonts • Decorations • Dialog Controls • Tab Controls • Menus • Automatic Resizing • Boolean Cluster Menus
CLD-VPP-20	Control front panel objects using Property Nodes and Control References		

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-20-01	Describe the attributes associated with Property Nodes <ul style="list-style-type: none"> • Read / Write • Adding Terminals • Execution Order • Cluster Inputs
		CLD-VPP-20-02	Explain common properties, including: <ul style="list-style-type: none"> • Visible • Disabled • Key Focus • Blinking • Value • Position • Bounds • Numeric Properties • Strings [4] (Boolean) • Display Style (strings)
		CLD-VPP-20-03	Explain common Graph and Chart properties, including: <ul style="list-style-type: none"> • X (or Y) Range • Active Plot • Plot Color • Active Cursor • Cursor Position • Cursor Index • Plot Area Size
		CLD-VPP-20-04	Given a set of requirements, create a VI that incorporates Property Nodes

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-VPP-20-05	Describe the function of a Control Reference
		CLD-VPP-20-06	Given a set of requirements, create a VI that incorporates Control References
CLD-VPP-21	Create Run-Time Menus in LabVIEW		
		CLD-VPP-21-01	Discuss the Run-Time Menu capabilities of LabVIEW, including <ul style="list-style-type: none"> • Static Menus • Dynamic Menus
		CLD-VPP-21-02	Describe the following items associated with LabVIEW Run-Time Menus <ul style="list-style-type: none"> • User Items • Application Items • Separators
		CLD-VPP-21-03	Demonstrate the use of the LabVIEW Menu Editor
		CLD-VPP-21-04	Describe the functions, inputs and outputs of the VIs associated with Menus: <ul style="list-style-type: none"> • Insert Menu Items • Delete Menu Items • Get Menu Item • Set Menu Item • Get Menu Shortcut Info
		CLD-VPP-21-05	Describe the functions, inputs and outputs of the VIs

Tasks and Objectives:
Certified LabVIEW Developer

			associated with Menu Selection Handling: <ul style="list-style-type: none"> • Current VIs Menubar • Get Menu Selection • Enable Menu Tracking
CLD-VPP-22	Create VIs with local and global variables		
		CLD-VPP-22-01	State the function of a: <ul style="list-style-type: none"> ▪ Local Variable ▪ Global Variable
		CLD-VPP-22-02	List the advantages and disadvantages of Local and Global Variables
		CLD-VPP-22-03	Explain the processes for creating Local and Global Variables
		CLD-VPP-22-04	Define the following terms associated with Variables: <ul style="list-style-type: none"> • Read Local / Global • Write Local / Global
		CLD-VPP-22-05	Explain the concept of a “race condition” and how to avoid race conditions when using Local Variables
		CLD-VPP-22-06	Describe initialized and un-initialized variables and their effect on VI operation
		CLD-VPP-22-07	Explain why owned labels are important in the use of Variables
		CLD-VPP-22-08	Discuss how the use of variables affects the dataflow

Tasks and Objectives:
Certified LabVIEW Developer

			programming concept and when variables should be used
CLD-VPP-23	Transmit data using datsocket		
		CLD-VPP-23-01	Describe the use of the following Datasocket VIs <ul style="list-style-type: none"> ▪ Datasocket Read ▪ Datasocket Write ▪ Datasocket Select URL
		CLD-VPP-23-01	Given a set of networked computers, generate a LabVIEW VI to transmit data, receive data, and display data using Datasocket VIs
CLD-VPP-24	Perform Byte Stream File I/O		
		CLD-VPP-24-01	List the differences between the text, Byte Stream, and Datalog file types
		CLD-VPP-24-02	Evaluate the use of using text, Byte Stream, and Datalog file types for a given set of needs and conditions
		CLD-VPP-24-03	For a given data set, describe the makeup of the header for a Byte Stream file
		CLD-VPP-24-04	Describe the use, inputs and outputs of the following functions for Byte Stream file I/O <ul style="list-style-type: none"> ▪ Open/Create/Replace VI ▪ File Dialog VI ▪ New File VI ▪ Open File VI

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Read File VI ▪ Write File VI ▪ Close File VI
		CLD-VPP-24-05	Given a set of requirements, create a Byte Stream file I/O VI
		CLD-VPP-24-06	Describe the concept of random access file I/O
		CLD-VPP-24-07	<p>Explain the use and requirements for the following inputs when performing random access file I/O</p> <ul style="list-style-type: none"> ▪ Pos offset ▪ Count ▪ # of values to read
		CLD-VPP-24-08	Given a set of requirements, create a Random access Byte Stream file I/O VI
		CLD-VPP-24-09	Describe the concept of streaming data to disk
		CLD-VPP-24-10	Given a set of requirements, create a Byte Stream file I/O VI that streams data to disk
CLD-VPP-25	Perform Datalog File I/O		
		CLD-VPP-25-01	<p>Describe the use, inputs and outputs of the following functions for Datalog file I/O</p> <ul style="list-style-type: none"> ▪ File Dialog VI ▪ New File VI ▪ Open File VI ▪ Read File VI ▪ Write File VI

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Close File VI
		CLD-VPP-25-02	Given a set of requirements, create a Datalog file I/O VI
CLD-VPP-26	Manage Projects in LabVIEW		
		CLD-VPP-26-01	Describe the use of the VI Revision History capabilities of LabVIEW
		CLD-VPP-26-02	Identify differences between two VIs using the VI Comparison capability in LabVIEW
CLD-VPP-27	Create stand-alone executable programs from VIs		
		CLD-VPP-27-01	Describe the process for building a stand-alone executable from a development environment VI
		CLD-VPP-27-02	Discuss the issues that affect the final built application <ul style="list-style-type: none"> ▪ Architecture of the application ▪ Programming issues ▪ Installer issues
		CLD-VPP-27-03	Given a set of conditions and requirements, build, install, and run an executable application using the LabVIEW Application Builder
CLD-AVP-01	Incorporate Accepted Programming and User Interface Design Techniques		

Tasks and Objectives:
Certified LabVIEW Developer

	into the Application Development Process		
		CLD-AVP-01-01	List and discuss the organization techniques inherent in a properly designed application <ul style="list-style-type: none"> ▪ Hierarchical file system ▪ Use of folders for grouping like functions ▪ Descriptive file names
		CLD-AVP-01-02	Evaluate Front Panel visual style features for ease of use <ul style="list-style-type: none"> ▪ Fonts and text styles ▪ Colors ▪ Imported graphics ▪ Use of Property nodes
		CLD-AVP-01-03	Evaluate Front Panel features relating to defining end-user input <ul style="list-style-type: none"> ▪ Use of Ring Controls ▪ Setting default values ▪ Defining ranges ▪ Key Navigation
		CLD-AVP-01-04	Evaluate Front Panel features relating to textual information <ul style="list-style-type: none"> ▪ Tip Strips ▪ Labeling
		CLD-AVP-01-05	Evaluate Front Panel features relating to logical grouping of related objects <ul style="list-style-type: none"> ▪ Logical Layout of objects ▪ Size and positioning ▪ Dialog Boxes

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-AVP-01-06	<p>State the advantages of incorporating good programming style in Block Diagram development</p> <ul style="list-style-type: none"> ▪ Easier to read and follow ▪ Easier to modify ▪ Professional look and feel
		CLD-AVP-01-07	<p>Create Block Diagrams that incorporate good wiring style</p> <ul style="list-style-type: none"> ▪ Avoid wiring underneath objects ▪ Minimize bends in wires ▪ Evenly space wires and objects ▪ Delete extraneous wires ▪ Avoid variables where wires can be used ▪ Do not pass wires through structures unless the data is used within that structure
		CLD-AVP-01-08	<p>Explain Block Diagram features that enhance readability</p> <ul style="list-style-type: none"> ▪ Size of the Block Diagram ▪ Left-to-Right Layout ▪ Block Diagram comments ▪ Omit labels on obvious functions ▪ Use labels to describe functionality of structures ▪ Use labels on wires to describe the data ▪ Comment the function of algorithms
		CLD-AVP-01-09	<p>Describe the characteristics of Block Diagram creation that enhance the speed of a VI</p> <ul style="list-style-type: none"> ▪ Time delays in loops ▪ Proper array building techniques ▪ Minimize the use of variables ▪ Avoid coercion dots

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Consider indicator speed
		CLD-AVP-01-10	<p>Describe the characteristics of Icons that add to ease of use</p> <ul style="list-style-type: none"> ▪ Create meaningful icons ▪ Always create black and white versions of icons ▪ Create standard size icons ▪ Use appropriate fonts for text
		CLD-AVP-01-11	<p>Describe the characteristics of Connector Panes that add to ease of use</p> <ul style="list-style-type: none"> ▪ Select connectors with extra terminals ▪ Select the same pattern for VIs with similar functions ▪ Connect inputs on the left and outputs on the right ▪ Use the required/recommended/optional selection for terminals ▪ Select patterns with sixteen or fewer terminals ▪ Relate connector pane layout to the Front Panel Layout if possible
CLD-AVP-02	Maximize Performance of VIs		
		CLD-AVP-02-01	<p>Define the following terms:</p> <ul style="list-style-type: none"> ▪ Cooperative Multitasking ▪ Preemptive Multitasking ▪ Multithreading
		CLD-AVP-02-02	State the benefits of preemptive multitasking
		CLD-AVP-02-03	Explain the conditions where multithreading should be avoided

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-AVP-02-04	State the purpose of the execution system and priority VI Properties
		CLD-AVP-02-05	Discuss the problems that can be introduced by improper selection of execution system and priority
		CLD-AVP-02-06	List the functions, VIs and operations that must run in the user interface thread <ul style="list-style-type: none"> ▪ Property Nodes ▪ Menu Control VIs ▪ VI Server ▪ Thread-unsafe CINI and DLLs ▪ Code Debugging ▪ Dialog Boxes
		CLD-AVP-02-07	Given a set of conditions and requirements, Set up a VI to use multiple threads for increased performance
		CLD-AVP-02-08	Discuss methods for troubleshooting performance problems <ul style="list-style-type: none"> ▪ Disabling Run in Multiple Threads ▪ Thread-unsafe CINI and DLLs ▪ Priority and execution system related problems ▪ Enabling Synchronous Display
CLD-AVP-03	Perform VI profiling		
		CLD-AVP-03-01	Describe the two main types of statistics gathered by the Profile window <ul style="list-style-type: none"> ▪ Execution Time Statistics

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Memory Statistics
		CLD-AVP-03-02	Explain the process for gathering information using the Profile window
		CLD-AVP-03-03	Given a set of conditions and requirements, use the Profile window to gather statistical information on a running VI
CLD-AVP-04	Optimize VI speed		
		CLD-AVP-04-01	Describe the effects of the following factors on VI execution speed <ul style="list-style-type: none"> ▪ Input/Output ▪ Screen Display ▪ SubVI Overhead ▪ Computation in Loops ▪ Use of Local and Global variables
		CLD-AVP-04-02	Given a set of conditions and requirements, modify a VI to increase the speed of execution
CLD-AVP-05	Optimize VI Memory Usage		
		CLD-AVP-05-01	List the factors that affect memory use <ul style="list-style-type: none"> ▪ Assembling Arrays ▪ Inplaceness ▪ Coercion ▪ Consistent Data Types ▪ Data Structures ▪ Global variables
		CLD-AVP-05-02	Discuss good programming techniques that aid in reducing

Tasks and Objectives:
Certified LabVIEW Developer

			memory use <ul style="list-style-type: none"> ▪ Assembling Arrays ▪ Inplaceness ▪ Coercion ▪ Consistent Data Types ▪ Data Structures ▪ Global variables
		CLD-AVP-05-03	Given a set of conditions and requirements, optimize the memory use of a VI
CLD-AVP-06	Manage Data Exchange		
		CLD-AVP-06-01	Explain the method of using uninitialized shift registers for data exchange
		CLD-AVP-06-02	Explain the method of using Queues for data exchange
		CLD-AVP-06-03	Explain the method of using Notifiers for data exchange
		CLD-AVP-06-04	Given a set of conditions and requirements, create a set of VIs that use an uninitialized shift register for data exchange
		CLD-AVP-06-05	Given a set of conditions and requirements, create a set of VIs that use Notifiers for data exchange
		CLD-AVP-06-06	Given a set of conditions and requirements, create a set of VIs that use Queues for data exchange
CLD-AVP-07	Synchronize Program Flow		
		CLD-AVP-07-01	Describe the concept of task synchronization

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-AVP-07-02	<p>Compare the following methods of synchronization, including any advantages or disadvantages</p> <ul style="list-style-type: none"> ▪ Global Variables ▪ Rendezvous ▪ Semaphores ▪ Occurrences
		CLD-AVP-07-03	<p>Given a set of conditions and requirements, create a VI that uses Global Variables for synchronization</p>
		CLD-AVP-07-04	<p>Explain the function, inputs and outputs of the Occurrence VIs</p> <ul style="list-style-type: none"> ▪ Generate Occurrence ▪ Wait on Occurrence ▪ Set Occurrence
		CLD-AVP-07-05	<p>Given a set of conditions and requirements, create a VI that uses Occurrences for synchronization</p>
		CLD-AVP-07-06	<p>Explain the function, inputs and outputs of the Semaphore VIs</p> <ul style="list-style-type: none"> ▪ Create Semaphore ▪ Acquire Semaphore ▪ Release Semaphore ▪ Destroy Semaphore
		CLD-AVP-07-07	<p>Given a set of conditions and requirements, create a VI that uses Semaphores for synchronization</p>
		CLD-AVP-07-08	<p>Explain the function, inputs and outputs of the Rendezvous VIs</p>

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Create Rendezvous ▪ Wait at Rendezvous ▪ Resize Rendezvous ▪ Destroy Rendezvous ▪ Get Rendezvous Status
		CLD-AVP-07-09	Given a set of conditions and requirements, create a VI that uses Rendezvous for synchronization
CLD-AVP-08	Communicate using TCP/IP		
		CLD-AVP-08-01	Explain the concept of TCP/IP communications
		CLD-AVP-08-02	<p>List the four layers of TCP/IP and their Protocols</p> <ul style="list-style-type: none"> ▪ Application/Presentation/Session <ul style="list-style-type: none"> ○ Simple Mail Transfer Protocol (SMTP) ○ File Transfer Protocol (FTP) ○ Telnet ▪ Transport <ul style="list-style-type: none"> ○ Transmission Control Protocol ▪ Network <ul style="list-style-type: none"> ○ Internet Protocol (IP) ○ Address Resolution Protocol (ARP) ▪ Datalink (Hardware) <ul style="list-style-type: none"> ○ Ethernet ○
		CLD-AVP-08-03	Describe the Client/Server Model
		CLD-AVP-08-04	<p>List the four major communication steps LabVIEW takes in the Client Model</p> <ul style="list-style-type: none"> ▪ Opens a connection to the server

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Sends commands to the server ▪ Receives responses from the server ▪ Close the connection and report errors
		CLD-AVP-08-05	<p>List the five major communication steps LabVIEW takes in the Server Model</p> <ul style="list-style-type: none"> ▪ Initializes the server ▪ Waits for a connection ▪ Waits for a command ▪ Execute the command and return the results ▪ Close the connection
		CLD-AVP-08-06	<p>Explain the function, inputs and outputs of the TCP/IP VIs</p> <ul style="list-style-type: none"> ▪ TCP Open Connection ▪ TCP Read ▪ TCP Write ▪ TCP Close Connection ▪ TCP Create Listener ▪ TCP Wait on Listener ▪ String to IP ▪ IP to String
		CLD-AVP-08-07	<p>Given a set of conditions and requirements, create a VI that acts as a TCP Client</p>
		CLD-AVP-08-08	<p>Given a set of conditions and requirements, create a VI that acts as a TCP Server</p>
		CLD-AVP-08-09	<p>Describe the use of the built-in web server</p>

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-AVP-08-10	Given a set of conditions and requirements, create a VI that displays a static web page
		CLD-AVP-08-11	Given a set of conditions and requirements, create a VI that displays an animated web page
CLD-AVP-09	Communicate between VIs with Datasocket		
		CLD-AVP-09-01	Describe the function and use of Datasocket
		CLD-AVP-09-02	Describe the components of Datasocket <ul style="list-style-type: none"> ▪ Datasocket API ▪ Datasocket Server
		CLD-AVP-09-03	List and describe the four components of a Datasocket communication <ul style="list-style-type: none"> ▪ Server ▪ Publisher ▪ Subscriber ▪ Data Item
		CLD-AVP-09-04	Create a typical Datasocket Transfer Protocol URL and describe each of the URL components
		CLD-AVP-09-05	Explain the function, inputs and outputs of the Datasocket functions: <ul style="list-style-type: none"> ▪ Datasocket URL VI ▪ Datasocket Write VI ▪ Datasocket Read VI
		CLD-AVP-09-06	Create a Datasocket connection between two interconnected

Tasks and Objectives:
Certified LabVIEW Developer

			computers using Datasocket function VIs and pass data from one to the other over the connection
		CLD-AVP-09-07	<p>Explain the use of the Datasocket Server Manager, including</p> <ul style="list-style-type: none"> ▪ Server Settings ▪ Permission Groups
		CLD-AVP-09-08	<p>Discuss the use of Variants for Datasocket communications, including the following functions:</p> <ul style="list-style-type: none"> ▪ To variant VI ▪ Variant to Data VI ▪ Variant to Flattened String ▪ Flattened String to Variant ▪ Set Variant Attribute VI ▪ Get Variant Attribute VI ▪ Delete Variant Attribute
		CLD-AVP-09-09	Given a set of conditions and requirements, create a set of VIs that perform bi-directional Datasocket communications
CLD-AVP-10	Implement VI Server		
		CLD-AVP-10-01	<p>Describe the function and capabilities of the VI Server</p> <ul style="list-style-type: none"> ▪ Call VIs remotely ▪ Configure remote LabVIEW machines as servers ▪ Change the properties of multiple VIs ▪ Dynamic load of VIs ▪ Create plug-in functionality for VIs
		CLD-AVP-10-02	Explain the following terms associated with object-oriented programming (OOP)

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ Object Oriented Programming ▪ Class ▪ Methods ▪ Properties
		CLD-AVP-10-03	<p>Explain the two classes of objects</p> <ul style="list-style-type: none"> ▪ Application Object ▪ VI Object
		CLD-AVP-10-04	<p>Explain the Open Application Reference function and its associated operations</p> <ul style="list-style-type: none"> ▪ Application Methods ▪ Property Nodes
		CLD-AVP-10-05	<p>Explain the Open VI Reference function and its associated operations</p> <ul style="list-style-type: none"> ▪ VI Object Properties ▪ VI Object Methods
		CLD-AVP-10-06	<p>Explain the purpose of the Close Application or VI Reference function</p>
		CLD-AVP-10-07	<p>Define the Strictly Typed VI Refnum, its use and advantages</p>
		CLD-AVP-10-08	<p>Describe the behavior of a strictly-typed Refnum</p>
		CLD-AVP-10-09	<p>Discuss the following VI Server options :</p> <ul style="list-style-type: none"> ▪ VI Server: Configuration ▪ VI Server: Exported VIs ▪ VI Server: TCP/IP Access

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-AVP-10-10	Given a set of conditions and requirements, create VIs to implement the LabVIEW VI Server
CLD-AVP-11	Create and Use Shared Libraries		
		CLD-AVP-11-01	Describe the functions, advantages, and limitations of a Dynamic Link Library (DLL)
		CLD-AVP-11-02	List the items needed to call a function in a DLL <ul style="list-style-type: none"> ▪ Data Type ▪ Calling Convention ▪ Parameters ▪ Location of the DLL ▪ Thread-safe or thread-unsafe
		CLD-AVP-11-03	List the purpose, inputs and outputs of the Call Library Function
		CLD-AVP-11-04	Given a set of conditions and requirements, create a VI to call a DLL from LabVIEW
		CLD-AVP-11-05	Given a set of conditions and requirements, debug errors returned from the Call Library Function
		CLD-AVP-11-06	Describe the process for building a DLL
		CLD-AVP-11-07	List the conditions required for a DLL to be considered thread-safe <ul style="list-style-type: none"> ▪ No global storage

Tasks and Objectives:
Certified LabVIEW Developer

			<ul style="list-style-type: none"> ▪ No hardware access ▪ No calls to thread-unsafe functions ▪ No use of semaphores or mutexes to protect global resources ▪ Not called from a reentrant VI
		CLD-AVP-11-08	Given a set of conditions and requirements, create DLL from LabVIEW
CLD-AVP-12	Implement ActiveX functions in LabVIEW		
		CLD-AVP-12-01	<p>Define the following terms:</p> <ul style="list-style-type: none"> ▪ Object Linking and Embedding (OLE) ▪ ActiveX ▪ ActiveX Automation Server ▪ ActiveX Automation Client ▪ ActiveX Automation Object
		CLD-AVP-12-02	<p>Explain the function, inputs and outputs of the ActiveX automation Client VIs</p> <ul style="list-style-type: none"> ▪ Automation Open ▪ Automation Close ▪ Invoke Node ▪ Property Node ▪ To Variant ▪ Variant to Data
		CLD-AVP-12-03	Given a set of conditions and requirements, create an ActiveX Client VI
		CLD-AVP-12-04	Describe the capabilities of a LabVIEW ActiveX container

Tasks and Objectives:
Certified LabVIEW Developer

CLD-AVP-13	Trap Errors in LabVIEW		
		CLD-AVP-13-01	<p>List and discuss the common questions to determine the behavior of an application</p> <ul style="list-style-type: none"> ▪ What are the inputs and outputs of the program? ▪ Is the user able to interact with the application ▪ What other hardware, software, networks, or third party programs/DLLs does the application use? ▪ Do other applications run under the same conditions? (on the same computer or network etc.)
		CLD-AVP-13-02	<p>List and discuss the common questions to determine the state of an application</p> <ul style="list-style-type: none"> ▪ Do all of the variables have appropriate values? ▪ Are there hidden variables, and do they have appropriate values? ▪ Is the application executing the appropriate code segments? ▪ Does the system have the correct IP address, device number, etc.?
		CLD-AVP-13-03	<p>List and discuss the common circumstances that surround difficult to correct problems</p> <ul style="list-style-type: none"> ▪ Combinations of circumstances that cause the error ▪ When one resource is in a particular state ▪ A sequence of events that leads to the problem, where each individual step is within bounds ▪ Side effects from a previous change or fix
		CLD-AVP-13-04	Describe a memory leak

Tasks and Objectives:
Certified LabVIEW Developer

		CLD-AVP-13-05	Discuss the types of situations that lead to memory leaks
		CLD-AVP-13-06	List and discuss the following multithreading errors that may occur <ul style="list-style-type: none">▪ Deadlock▪ Priority Inversion▪ Thread Starvation