Certification ID	Certification Title	Job Description:
CLA	Certified LabVIEW Architect	Given a set of requirements for a large application, the Certified LabVIEW Architect is able to develop, lead, and direct a team of LabVIEW developers in the creation of an efficient, cost effective solution

Task ID	Task	Objective ID	Objective
CLA-ADT-01	Apply standard, accepted high-level design techniques to a LabVIEW application		
		CLA-ADT-01-01	Formulate a prioritized plan for developing a large LabVIEW application ▪ Appropriate customer / designer interaction ▪ Identify follow-up items, concerns, and outstanding issues ▪ Generate plans and estimates for the application
		CLA-ADT-01-02	Create design guidelines for a multi-developer LabVIEW application ▪ interaction between code segments ▪ data requirements

		CLA-ADT-01-03	Given a set of conditions, create use cases for an application in a general written format (no UML or pseudo-code required)
		CLA-ADT-01-04	Create a set of clear, concise, testable software requirements for an application
		CLA-ADT-01-05	Create a test plan to implement testing requirements
		CLA-ADT-01-06	Describe the benefits of using programming constructs in creating scalable, maintainable applications
CLA-ADT-02	Produce a LabVIEW application using appropriate programming style for a team environment		

		CLA-ADT-02-01	Describe proper style, coding standards and issues relating to individual and team development, including: ▪ VI naming conventions ▪ Source Code Control ▪ SubVI Integration ▪ Consistent UIs ▪ Scalability ▪ Documentation ▪ Hard coded paths, names, or associated information ▪ Excessive use of variables ▪ Overly large diagrams ▪ Use of appropriate data types ▪ Labeling wires from shift registers ▪ Appropriate use of sequence structures and sequence locals
		CLA-ADT-02-02	Generate templates that can be used in a team environment to encourage suitable programming style
CLA-ADT-03	Perform code reviews on LabVIEW applications		
		CLA-ADT-03-01	Describe the importance of proper documentation as applicable to team development and code maintenance

Revision December 2, 2002

| | | CLA-ADT-03-02 | Given a set of conditions, review a LabVIEW application for overall design, including:
▪ Documentation
▪ Structures labeled
▪ Description of VI functionality
▪ Sufficient level of detail
▪ Self-documenting code
▪ Efficiency
▪ Coding Style
▪ Readability
▪ Appropriate Front Panel Design |
| | | CLA-ADT-03-03 | Given a set of conditions, evaluate a LabVIEW application for programming considerations, including:
▪ Effectiveness of Algorithms
▪ Proper use of Dataflow programming techniques
▪ Efficient Data / Memory management
▪ Common coding defects including:
　・ Open References
　・ Invalid references
　・ Potential race conditions |

Revision December 2, 2002

		CLD-VPP-03-04	Review a LabVIEW application for run-time behavior and memory management issues, including: - Inplaceness - Use of Refnums vs. values - Use of property nodes vs. values - Wire Branching - Execute and operate data - Storage of data types - Screen updates - Moving large amounts of memory - Synchronous and asynchronous updating - Transfer buffers - Execution order - Destructive and non-destructive buffer reads on branched wires
CLA-ADT-04	Utilize Source Code Control		
		CLA-ADT-04-01	Describe the need for source code control in a multi-developer environment
		CLA-ADT-04-02	Discuss the link between source code control and project management
		CLA-ADT-04-03	Describe the pitfalls that can be encountered when implementing source code control for a project

		CLA-ADT-04-04	Describe VI cross-linking and its effect on source code control
		CLA-ADT-04-05	Explain the impact of source code control on the release of an application
CLA-ADT-05	Design and Develop object-oriented code		
		CLA-ADT-05-01	Describe the conditions under which object-oriented coding techniques would be appropriate for a LabVIEW application
		CLA-ADT-05-02	Explain the design and implementation considerations necessary for creating an object-oriented application
		CLA-ADT-05-03	Incorporate functional globals appropriately in an application
		CLA-ADT-05-04	Describe the use of semaphores and synchronization techniques
		CLA-ADT-05-05	Describe encapsulation of data and functionality
		CLA-ADT-05-06	Explain the methods for developing scalable applications
		CLA-ADT-05-07	Create Class VIs and Type Definitions ▪ Single Enum Datalog Refnums used to identify Class Objects

CLA-ADT-06	Develop LabVIEW applications using plug-in techniques		
		CLA-ADT-06-01	Explain the use of the VI Server Call By Reference Node
		CLA-ADT-06-02	Describe the method by which VI Server is used to call a VI via ▪ Run Method ▪ Call By Reference Node
		CLA-ADT-06-03	Discuss the advantages and disadvantages of using a plug-in architecture in a LabVIEW application
		CLA-ADT-06-04	Discuss the advantages and disadvantages of using the Run Method versus using Call by Reference when creating a LabVIEW application
		CLA-ADT-06-05	Explain the process for setting input values using VI Server for using Run Method
		CLA-ADT-06-06	Generate standard VIs that call plug-in VIs
		CLA-ADT-06-07	Design a VI interface that properly identifies VIs as matching plug-ins
CLA-ADT-07	Develop LabVIEW Applications using messaging architectures		

		CLA-ADT-07-01	Describe the reasons for creating a User Interface that is separate from the top-level application VI
		CLA-ADT-07-02	Create a LabVIEW application with a User Interface that is separate from the top-level VI ▪ Call the user interface dynamically ▪ Communicate with the user Interface via common messaging architectures
		CLA-ADT-07-03	Explain the use and proper application of control Refnums and control Property Nodes
		CLA-ADT-07-04	Create an application that uses control Refnums and control property nodes to update controls in other VIs
		CLA-ADT-07-05	Generate an application that uses messaging architectures to pass data between parallel loops
		CLA-ADT-07-06	Develop reentrant VIs and describe the effects of reentrancy on debugging tools
		CLA-ADT-07-07	Explain the use of the common methods of communications between VIs: ▪ Queues ▪ Notifiers ▪ VI Server ▪ TCP/IP

Revision December 2, 2002

		CLA-ADT-07-08	Generate an application that executes multiple segments in parallel with communication via one of the following: ▪ Queues ▪ Notifiers ▪ VI Server ▪ TCP/IP
CLA-ADT-08	Create / incorporate modular code in LabVIEW		
		CLA-ADT-08-01	Describe the importance of backward compatibility with modular code
		CLA-ADT-08-02	Explain the implications and risks associated with multiple VIs that have the same names on a system
		CLA-ADT-08-03	Describe the implications of using project-specific global VIs
		CLA-ADT-08-04	Generate VIs using generic naming conventions
		CLA-ADT-08-05	Update a module while maintaining backward compatibility
		CLA-ADT-08-06	Categorize SubVIs for effective module organization
CLA-ADT-09	Deploy a LabVIEW Application		

Revision December 2, 2002

		CLA-ADT-09-01	Describe the design considerations related to deployment that must be observed when developing a large application
		CLA-ADT-09-02	Given a set of conditions, specify the items that must be deployed to a target machine
CLA-ADT-10	Optimize the use of execution systems and priorities for a LabVIEW application		
		CLA-ADT-10-01	Explain the proper use and pitfalls for misuse of: ▪ execution systems ▪ priority settings ▪ subroutine priority
		CLA-ADT-10-02	Optimize the performance of a LabVIEW application
CLA-ADT-11	Implement Proper Error Handling Techniques in LabVIEW applications		
		CLA-ADT-11-01	Describe important attributes of error handlers
		CLA-ADT-11-02	Explain where errors should be handled and reported

		CLA-ADT-11-03	Generate an applicable error handling arrangement for a large LabVIEW application: ▪ Generate and append appropriate additional information to the error handling system ▪ Filter errors as appropriate ▪ Handle or pass errors on as appropriate ▪ Logs errors if conditions are set to do so (basically have a debug flag)
CLA-ADT-12	Generate recursive code in LabVIEW applications		
		CLA-ADT-12-01	Describe recursion and circumstances when the use of recursion is appropriate
		CLA-ADT-12-02	Explain the use of VI Server in making recursive calls
			Describe the two general methods for creating recursive routines in LabVIEW, and the advantages and disadvantages of each ▪ Iterative-Recursive algorithm ▪ Recursive VI
		CLA-ADT-12-03	Given a set of requirements, generate a recursive routine in LabVIEW