CLAD 01 Sample Exam Answer Key

Question	Answer	Topic	Justification	LabVIEW Help Topic
1	D	LabVIEW Programming Principles	Functions and VIs execute as soon as all of their inputs have data. Thus, it is the flow of data through the program that determines the order of execution.	Block Diagram Data Flow
2	C	Charts and Graphs	Strip charts start plotting from left to right and continue to scroll while plotting. Scope charts start plotting from left to right and continue until the chart is full. Then the chart is cleared, and plotting resumes at the left. Sweep charts behave similarly to scope charts, except that once the chart is full, sweep charts start plotting at the left and progressively overwrite previously plotted data. There is no such thing as a Step chart in LabVIEW.	Customizing Graphs and Charts
3	B	Debugging	Code containing breakpoints executes like normal until data is passed on a wire that contains a breakpoint. Then, LabVIEW suspends execution. In order for the breakpoint to activate, the section of code containing it must be called. With a Case structure, only one case executes each time it is called. Since, in this case, the case containing the breakpoint was not called, the execution was not paused.	Managing Breakpoints
4	B	Design Patterns	A state machine can be used to achieve the same functionality as a sequence structure. However, state machines allow the developer to programmatically determine the sequence at run-time. This makes the state machine implementation more scalable.	Case and Sequence Structures
5	C	Loops	The While Loop has a FALSE Boolean wired to its conditional terminal, which is set to Stop if TRUE. Thus, there is no condition that causes this While Loop to stop. So, the While Loop could run infinitely and the VI must be aborted.	For Loop and While Loop Structures, While Loop
6	B	Local Variables	Local variables do not conform to the Dataflow paradigm because they communicate by reference, not by value. The basic premise of local variables is to allow transfer of data where it is impossible to use wires. This circumvents the Dataflow paradigm.	Block Diagram Data Flow, Local Variables, Using Local and Global Variables Carefully
7	B	Design Patterns	In LabVIEW, simple state machines consist of a While Loop containing a case structure. The case structure allows the decision of which case to run to be determined programmatically. It is also scalable since it is easy to add new cases to the case structure.	Creating VIs from Templates, Case Structure
8	C	Loops	An empty array is wired to the For Loop using an auto-indexing tunnel. This causes the For Loop to iterate once for every element in the array, which, in this case, is zero. However, the value 5 is written to the shift register before loop execution, and since the loop iterates zero times, the same value of 5 is present at the output shift register.	For Loop, Passing Multiple Values to the Next Loop Iteration, Using Shift Registers to Remember Iteration Values (Topic of the LabVIEW 2010 Help)
9	A, B, C, D	Property Nodes	All of the statements are true for Property Nodes.	Property Node
10	A	VI Server	Strict property nodes require precise knowledge of the numeric representation. Since the question asks for the means to change the text color of any control, a strict Property Node does not suffice. An implicit Property Node (B), can only be used locally. Answer C shows a property that is not even relevant to the question. Thus, option A is the best answer.	Switching Between Strictly Typed and Weakly Typed Control Refnums, Property Nodes

Question	Answer	Topic	Justification	LabVIEW Help Topic
11	B	Data Types	The integers shown are unsigned 8-bit integers. The range for 8-bit integers is 0-255. The product of 2 times 128 is 256, one more than the maximum allowable value for an unsigned 8-bit integer. Thus, the value wraps around to 0.	Numeric Data Types Table, Numeric Conversion
12	A	Debugging	Clicking the Step Into button causes LabVIEW to open up the node or subVI. The Step Out button is used to return from a subVI to a main VI while single-stepping. There is no Step Through button. Step Over provides the functionality stated by the question, therefore this is the correct answer.	Single-Stepping through a VI
13	D	Error Handling	Since automatic error handling is enabled in the main VI, and the error terminals of the subVI are not wired, LabVIEW automatically handles the error by displaying a dialog.	Handling Errors
14	D	LabVIEW Environment	Icons cannot be edited from the functions palette. SubVI icons can be edited by right- clicking the icon in the upper right of the VI and selecting Edit Icon.	Creating a VI Icon
15	B	Arrays and Clusters	The Initialize Array function creates an array with a length specified by dimension size. The value of each element is specified by the element input. To initialize multidimensional arrays, you can simply expand the Initialize Array function to display more dimension size inputs.	Initialize Array Function
16	D	VI Server	The only answer option that can be passed from calling VI to subVI is the control reference. The reference can then be used with Property Nodes and Invoke Nodes to call properties and methods, respectively. The data type is a property of the control.	Controlling Front Panel Objects Programmatically from a SubVI, VI Server Reference
17	C	Synchronization and Communication	Queues, notifiers, and local variables are all designed to transfer data. Semaphores do not pass data. Instead, their sole purpose is to prevent certain sections of code from running while other critical sections are running.	Synchronization VIs and Functions
18	D	Loops	The For Loop executes 5 times. Starting with the value of 1, the result of the previous iteration is multiplied by 2. Thus, the value in the indicator after 5 iterations is equivalent to 1x2x2x2x2x2 (25), or 32.	Passing Multiple Values to the Next Loop Iteration, Using Shift Registers to Remember Iteration Values (Topic of the LabVIEW 2010 Help)
19	A	LabVIEW Programming Principles	Because LabVIEW is a Dataflow language, we can trace the flow of data in the block diagram to see which operations execute first, second, and so forth.	Block Diagram Data Flow
20	C	Loops	For Loops refer to the input to the Count terminal to determine how many iterations to execute.	For Loop

Question	Answer	Topic	Justification	LabVIEW Help Topic
21	D	LabVIEW Environment	You must either look at an output error cluster or an error dialog to find the error code.	Error List Window
22	C	Mechanical Action of Booleans	Since the mechanical action is set to Switch Until Released, two events are generated when a user clicks and releases. The first event is the FALSE to TRUE transition, and the second is the TRUE to FALSE transition. Latch actions are specifically designed to reset the value of the button after the change is read without generating a second event.	Changing the Mechanical Action of a Boolean Object, Value Change Event, Using Events with Latched Boolean Controls
23	D	Event Structures	Each possible answer refers to a Value Change event. Value Change events are either generated by user interaction on the front panel, or by calling a Value (Signaling) Property Node. Calling a Value Property Node does not generate and event.	Available Events, Using Events in LabVIEW
24	D	File I/O	LabVIEW represents arrays as a list of bytes containing a header and array data. The header contains a 4-byte integer for each dimension that specifies the length of that dimension. Following the header is the actual array data. In the question, the array has two dimensions. There are 4 bytes in the header for each dimension for a total of 8 bytes in the header. Since the array is composed of 8-bit, or 1-byte integers, there is a total of 9 bytes of actual array data. This makes a total of 17 bytes being written to file.	Flattened Data, Creating Binary Files, How LabVIEW Stores Data in Memory
25	D	Case Structures	Arrays are not accepted by the case selector terminal because the case selector terminal requires a scalar value.	Creating Case Structures
26	A, C	Arrays and Clusters	Auto-indexing is a feature for loops interacting with arrays. Array functions themselves do not have iterative auto-indexing features.	For Loop and While Loop Structures, Passing Elements in an Array through a Loop, Enabling Auto-Indexing for Loops (Topic of the LabVIEW 2010 Help)
27	D	Functional Global Variables	You can place critical data or sections of code in functional global variables. Since functional global variables are non-reentrant VIs, the possibility of race conditions is eliminated.	Suggestions for Using Execution Systems and Priorities
28	D	Property Nodes	Since the indicator terminal is not isolated from the data being written by any structures, there is no reason not to wire the data directly to the indicator instead of using a Property Node. It is always best practice to wire directly when possible, because using variables or Property Nodes to update values can cause race conditions if not used carefully.	Block Diagram Data Flow, Using Local and Global Variables Carefully
29	A	Charts and Graphs	XY graphs accept a cluster of two arrays, an X array and a Y array to generate a single plot. To generate multiple plots, XY graphs accept an array of these clusters.	XY Graphs
30	B	Documentation	The documentation window in VI Properties is the only place to edit the information about the VI that appears in Context Help.	VI Description Property, Documentation Page (VI Properties Dialog Box)

Question	Answer	Topic	Justification	LabVIEW Help Topic
31	D	Timing	Answers B and D are incorrect because the Wait Until Next ms Multiple function executes before loop iterations terminate, not after. Answer C is incorrect, because it describes the functionality of the Wait (ms) function.	Wait Until Next ms Multiple Function
32	D	Timing	The Wait (ms) function does nothing to release or allocate memory or specify processor core. All it does is cause the execution of a VI to pause for a short time to allow the processor time to complete other tasks.	Wait (ms) Function
33	A	Arrays and Clusters	There is no need to make an array of arrays since you can simply add dimensions to an existing array.	Changing Array Dimensions
34	B	Arrays and Clusters	The Array Subset function takes and input array and returns a section of that array as specified, starting at the given index and continuing for a number of elements equal to length. Here, the index value 3 specifies the fourth element of the array, or 10. Since the specified length is value 4, an array of length 4 is returned as follows: {10, 8, 5, 7}.	Array Subset Function
35	C	Event Structures	The event case shown handles two events: the Mouse Down event on the Button control, and the Mouse Down Event on the pane, or front panel. These two events occur simultaneously, so the event structures queues them up, and handles one and then the other. Thus two iterations occur, and the value in the shift register is incremented twice.	Using Events in LabVIEW, Mouse Down Event
36	C	Sequence Structures	The sequence local is first written to in frame 1. Thus, in frame zero, no data is available to read since no data has yet been written.	Adding and Removing Sequence Local Terminals
37	B	Error Handling	The Merge Errors function does not display any dialog. The One and Two Button Dialog functions are for general purpose and are not best applied for error handling applications. There is no error input on these functions. There is no VI with the name Generate Front Panel Activity. The Simple Error Handler is the best choice because it accepts an error cluster as an input and displays a dialog to the user in the event of an error.	Handling Errors, Simple Error Handler VI
38	A	Synchronization and Communication	Answer A is incorrect because semaphores cannot pass data. Answer C is incorrect because notifiers pass data, but they can only pass one element at a time. Data is overwritten and lost if the program writes to the notifier twice before data is read. Answer B is incorrect because local variables have no mechanism for determining when data is updated, so there is no way to tell if data is newly-acquired or not. Queues support multiple elements and operate using a FIFO principle, guaranteeing that no data is lost or overwritten.	Synchronization VIs and Functions
39	D	LabVIEW Programming Principles	When possible, it is always best to wire data directly to indicators. This helps maintain the Dataflow paradigm.	Block Diagram Data Flow, Using Local and Global Variables Carefully
40	C	Data Types	Rings can use any numeric representation while enums can use only unsigned integers.	Ring Constant, Enum Constant