Question	Answer	Topic	Justification
1	D	LabVIEW Programming Principles	Utilizing error clusters in subVIs is the best way of enforcing data flow since it also allows for proper error handling.
2	B	Error Handling	Answers C and D report the error, but do not stop. Answer A stops but does not report the error. Answer B is the only acceptable option.
3	B	LabVIEW Programming Principles	For loops to run in parallel, there needs to be no data dependency between them. A wire running from one loop to another makes one loop dependent on the first due to the rules of data flow. Thus, if a wire is used between the loops, they are no longer in parallel.
4	D	Local Variables	A functional global variable will not work since it does not directly have a means for manipulating a front panel control. Data Value References do not apply in this situation. Setting the desired value as default is not a programmatic means of updating the control. A Local Variable is the only feasible answer.
5	B	Synchronization and Communication	The main benefit of global variables is their ability to pass data between multiple VIs.
6	B	Property Nodes	Property Nodes always execute from top to bottom. Plot 1 is selected and modified first so it will change color first.
7	D	Charts and Graphs	Waveform Graphs either accept 2D arrays or a 1D array of clusters containing waveform data. Waveform Graphs do not accept X and Y data interleaved in a 1D array as answer A suggests.
8	D	Event Structures	Notify events simply inform LabVIEW that an event has occurred and been processed. Filter events allow the user to programmatically decide to discard events.
9	B	LabVIEW Programming Principles	Since the .ctl file was saved as a Control and not a Type Def. or a Strict Type Def., the change to the file does not update instances of the control.
10	D	Synchronization and Communication	By default, the Dequeue Element function waits until data is available. It returns when data becomes available or the queue reference is destroyed. This functionality is important to the Producer/Consumer design pattern.
11	C	Sequence Structures	Sequence locals store data between frames of Stacked Sequence structures. The sequence local is only written to in frame 0. Thus frame 1 has no impact on frame 2. The value in Result F2 is 8 times 5, which is 40.
12	B	Loops	Stacked shift registers remember values written for multiple previous iterations. The top node is the most recent iteration. By looking at the code snippet, it is apparent that four most recent measurements are being averaged and displayed.
13	A	Debugging	The Probe tool allows the developer to see data in a wire during execution, but it does not slow down execution at all. Highlight Execution slows down execution and displays the flow of data, and the Single Stepping tools allow the developer to look into subVIs.
14	D	Case Structures	The Default case executes because the input to the case selector does not fit any of the other cases. Thus, the answer is 62, or 36.
15	A	Debugging	Breakpoints cause the VI to pause execution and wait for the developer to decide to start single-stepping or to unpause the execution.
16	B	Arrays and Clusters	When doing array arithmetic, LabVIEW will force the output to be the size of the smaller input. In this case, the output will be a 1D array with two elements. The elements are 75-100 and 50-25, or {-25, 25}.
17	D	Mechanical Action of Booleans	Windows dialog buttons wait until a user releases before processing the click. When the user clicks and releases, the button returns to its default state. This behavior is similar to the Latch When Released mechanical action in LabVIEW.

1	D	LabVIEW Programming Principles	Utilizing error clusters in subVIs is the best way of enforcing data flow since it also allows for proper error handling.
18	D	Functional Global Variables	It is not necessary to inline functional global variables into their calling VIs. In fact, inlining requires that the subVI be reentrant, which is forbidden for functional global variables.
19	B	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.
20	D	Design Patterns	State machines consist of a While Loop, a Case structure, a shift register, and code for determining the transition.
21	A	Arrays and Clusters	Clusters allow grouping of data into structures. This cleans up block diagrams by minimizing the number of wires and terminals required. Data types may be mixed in clusters.
22	B	Data Types	Coercion dots indicate that a certain data type is being wired to terminal that accepts a different but compatible data type. When this happens, LabVIEW converts the data to the larger of the two data types. This requires the creation of a memory buffer to store the coerced data.
23	D	Property Nodes	Implicit Property Nodes are explicitly linked to their owning control or indicator. No reference wires are needed. Explicit Property Nodes require a reference wire to determine which control the Property Node is manipulating. Data Value References have nothing to do with Property Nodes.
24	D	Design Patterns	A simple state machine has no buffer for maintaining state transitions that may have occurred during the execution of the previous state, such as a user clicking a front panel button. Thus, if more than one transition occurs before they can be processed, all but the most recent one will be lost.
25	B	Loops	The count terminal indicates how many times a For Loop will execute. The Conditional Terminal for a While Loop may determine how many times the loop executes, but there is no way to directly tell how many iterations will occur. The iteration terminal returns the number of iterations that have occurred, minus one.
26	B	LabVIEW Environment	LabVIEW Projects do not own files like folders do. Projects simply organize files that are part of a common application.
27	C	Loops	The iteration terminal in While Loops and For Loops always starts counting at zero. It returns 0 on the first iteration, 1 on the second iteration, etc. Since the While Loop is configured to stop when the output of the iteration terminal is greater than or equal to 50, we know that the iteration terminal must output a value of at least 50. The first time this happens is after 51 iterations.
28	A	Event Structures	Each possible answer refers to a Value Change event. Value Change events are detectable by an Event Structure when generated from user interaction with the front panel.
29	B	Timing	The Tick Count (ms) function returns the value of the millisecond timer when it is called. Calling it twice and taking the difference will yield elapsed time in milliseconds between the calls. In this code snippet, SubVI is called between the two calls of the Tick Count (ms) function, therefore A is the correct answer.
30	C	VI Server	Property Nodes are designed for modifying front panel objects programmatically. Answers A and B do not make sense because modifying front panel objects is not what variables do. Because we are not using subVIs, answer C is a better choice than answer D because an implicit Property Node will work and does not require the extra inputs that the explicit Property Node requires.
31	B	Arrays and Clusters	For Loops are more efficient at creating arrays than While Loops because For Loops execute for a predetermined number of iterations. Thus, LabVIEW can allocate the memory to be used by the array before the For Loop runs.
32	B	Timing	Answers C and D both return values in milliseconds. Since the question specifies applications that run for extended periods of time, we should choose a function with a larger resolution to minimize the likelihood that the clock value will wrap around to zero. Furthermore, answer A simly adds a synchronous delay to code. This would not be efficient for applications running for long periods of time. It is better to compare the current time to a reference time to see if the correct amount of time has elapsed. Thus, answer B is the best choice.
33	D	Error Handling	Even though automatic error handling is enabled, all of the functions have error clusters wired to their outputs. This prevents LabVIEW from interrupting execution and displaying a dialog to the user even though an error has occurred. If any of the functions had no error cluster wired to their outputs, then LabVIEW would display a dialog and suspend execution.
34	B, D	Documentation	A bold input in the Context Help window indicates a Required input. Because of the nature of Required inputs, the VI will have a broken run arrow if the input is unwired.

1	D	LabVIEW Programming Principles	Utilizing error clusters in subVIs is the best way of enforcing data flow since it also allows for proper error handling.
35	B	Arrays and Clusters	When given a multidimensional array, the Array Size function will output a 1D array containing the size of each dimension. The order of arrays is always row first, column second. Thus, the correct answer is B since there are 2 rows and 3 columns.
36	A	File I/O	A single Boolean value uses one byte of memory in LabVIEW. Answer A is the correct answer because the Boolean Array to Number converts 8 Boolean values, or 8 bytes, into a single 8-bit integer value, or 1 Byte. Thus the total amount of data written is 1 byte. Answer B writes 1 byte for each Boolean value, and answers C and D write strings. Each character in a string uses 1 byte, so answers C and D each write multiple bytes of data to the file.
37	A	LabVIEW Environment	Key Navigation is a property of controls that allows the programmer to assign keyboard actions to controls. Key Focus is a property that determines if a control is currently selected or not. A radix is a display component of integer controls which allows the user to select between decimal, binary, octal, and hexadecimal display. Distribute Objects is a tool for organizing the front panel.
38	D	Data Types	For a given string input, the Replace Substring function replaces the original string starting at the location specified in offset with the string supplied in the substring input.
39	D	VI Server	By passing control references to other VIs, programmers allow SubVIs access to the properties and methods of objects in the main VI.
40	C	Loops	Feedback nodes function very similarly to shift registers. In this code snippet, the feedback node is initialized with a value of 4. It iterates once and the value 4 is added to the value stored in the feedback node, giving a resulting value of 8.