Exam 3: Learning Objectives under Evaluation

Exam 2 will cover:		
LOs 01.00 – 09.00 (Exam 1)	14.00 Relational/Logical Operations	17.00 Repetition Structures
LOs 10.00 – 13.00 (Exam 2)	15.00 Flowcharts	18.00 Indexed Looping
	16.00 Selection Structures	19.00 Nested Structures

14.00 Perform and evaluate relational and logical operations

Learning Objective	Evidence
14.01 Construct relational and logical statements from English statements	Identify correct true/false pattern for the English statement Logical statement is correct Complete work is shown when verifying that the logical statement results in the pattern identified Logical statement is only as complicated as necessary
14.02 Employ relational operators with arrays (scalars, vectors, matrices)	Variables or values are placed correctly within the statement(s) Complete work is shown when evaluating expressions Arrays must have the same dimensions (unless one is a scalar) Correct syntax for relational operators (>, >=, <, <=, ==, ~=) Evaluate or use appropriate relational operators to compare arrays Apply output of relational operation as logical true/false
14.03 Employ order of operations to perform calculations, comparisons, and logical operations	Variables or values are placed correctly within the statement(s) Complete work is shown when evaluating expressions Apply the MATLAB order of precedence: 1. Parentheses () 2. Transpose (.'), power (.^), complex conjugate transpose ('), matrix power (^) 3. Logical negation (~) 4. Multiplication (.*), right division (./), left division (.\), matrix multiplication (*), matrix right division (/), matrix left division (\) 5. Addition (+), subtraction (-) 6. Less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (==), not equal to (~=) 7. Element-wise AND (&) 8. Element-wise OR (|) 9. Short-circuit AND (&&) 10. Short-circuit OR (||) In the absence of parentheses, operators at the same level are read left to right (with some exceptions that are not discussed in ENGR 132, see

	https://www.mathworks.com/help/matlab/matlab_prog/operator-precedence.html) The mathematical expression a<x<b be coded as a<x & x<b
14.04 Employ comparison functions with vectors and matrices: any, all	Correct syntax for any command: any(X) Correct syntax for any command: all(X) Use and evaluate vectors or matrices with zero and non-zero value elements within the any command • any(vector) operates on the full vector and returns a logical scalar of 1 if a single value is a non-zero value • any(matrix) operates on the columns of matrix and returns a corresponding vector of logical elements where the values of 1 indicate columns with one or more non-zeros values Use and evaluate vectors or matrices with zero and non-zero value elements within the all command • all(vector) operates on the full vector and returns a logical scalar of 1 if every value is a non-zero value • all(matrix) operates on the columns of matrix and returns a corresponding vector of logical elements where the values of 1 indicate columns with all non-zeros values Differentiate between appropriate use of any or all Apply output from any or all command as logical true/false
14.05 Employ comparison functions with vectors and matrices: find	Correct syntax for find(x) Evaluate a call to find Recognize that the output of find is an array of indices for which the logic statement is true Apply output of find as indices
14.06 Employ logical operations with arrays (scalars, vectors, matrices)	Variables or values are placed correctly within the statement(s) Complete work is shown when evaluating expressions Arrays have the same dimension (unless one is a scalar) Correct syntax for logical operators (&, |, ~) Correct syntax for xor command: xor(X) Evaluate or use appropriate logical operators with arrays • Exclusive or (xor) returns true when only one element is true • Inclusive or (|) returns true when one or both elements are true • AND (&) returns true when both elements are true Example: The mathematical expression a<x<b is coded as a<x & x<b • Logical negation (~) reverse the output (true to false, false to true)

Exam 3: Learning Objectives under Evaluation

	Apply output of logical operation as logical true/false
14.07 Construct truth tables to evaluate logical expressions	When there are two variables: • a 2x2 table is used • 2 columns for variable 1 (T/F) • 2 rows for variable 2 (T/F) When there are three variables: • a 2x4 (or 4x2) table is used • 4 columns for an orderly presentation of all combinations of variable 1 (T/F) and variable 2 (T/F) • 2 rows for variable 3 (T/F) When there are four variables: • a 4x4 table is used • 4 columns for an orderly presentation of all combinations of variable 1 (T/F) and variable 2 (T/F) • 4 rows for an orderly presentation of all combinations of variable 3 (T/F) and variable 4 (T/F)

Exam 3: Learning Objectives under Evaluation

15.00 Construct and troubleshoot a flowchart using standard symbols and pseudocode

Learning Objective	Evidence
15.01 Construct a flowchart for a selection structure using standard symbols and pseudocode	Decisions that are part of a selection structure are represented with a diamond filled with a condition Decision have one input arrow and two output arrows (one for Yes/True and one for No/False) There are operations on the Yes/True path For a single selection (i.e., if-end), there are no operations on the No/False path before the convergence of the Yes/True and No/False path arrows For two possible selections (i.e., if-else-end), there are operations on the No/False paths before the convergence of the Yes/True and No/False path arrows For multiple related selections (i.e., if-elseif-else), there are no operations between the decisions along the No/False path For multiple related selections (i.e., if-elseif-else), the Yes/True and No/False path arrows converge after all related decisions and (optionally) the operations for the else path Operations are included in the selection structure as required by the problem
15.02 Track a flowchart with a selection structure	Identify correct path given the test value(s) Describe the outcome(s) in English with resulting values when appropriate (not code results)
15.03 Construct a flowchart for an indefinite looping structure using standard symbols and pseudocode	A decision indicating an indefinite loop is represented by a diamond filled with a condition A decisions indicating an indefinite loop has one input arrow from processes prior to the loop, one input arrow from processes inside the loop, and two output arrows representing the Yes/No or True/False paths The Yes/True path enters the indefinite loop The No/False path exits the indefinite loop All variables used in the decision or in recursive calculations within the indefinite loop are initialized before the decision All variables used in the decision are updated within the indefinite loop Operations are included in the indefinite loop as required by the problem
15.04 Track a flowchart with an indefinite looping structure	Describe the overall purpose of the indefinite loop in English Describe how the indefinite loop initiates and terminates Determine the correct number of iterations Determine the value(s) of all variable(s) (re)assigned in the loop prior to the start of the loop (iteration 0) Determine the value(s) of all variable(s) (re)assigned in the loop at the end of each iteration Determine the value(s) of all variable(s) (re)assigned in the loop at the end of loop execution
15.05 Construct a flowchart for a definite looping structure using standard symbols and pseudocode	A decisions indicating a definite loop is represented by a diamond broken into three parts indicating the loop index start value, the index value condition to terminate the loop, and the index increment A decision indicating a definite loop has one input arrow from processes prior to the loop, one input arrow from processes inside the loop, and two output arrows representing the Yes/No or True/False paths The Yes/True path enters the definite loop The No/False path exits the definite loop

Exam 3: Learning Objectives under Evaluation

	All variables that used in recursive calculations within the definite loop are initialized before the decision Operations are included in the definite loop as required by the problem
15.06 Track a flowchart with a definite looping structure	Describe the overall purpose of the definite loop in English Describe how the definite loop initiates and terminates Determine the correct number of iterations Determine the value(s) of all variable(s) (re)assigned in the loop prior to the start of the loop (iteration 0) Keep track of the value of the definite loop index at each iteration Determine the value(s) of all variable(s) (re)assigned in the loop at the end of each iteration Determine the value(s) of all variable(s) (re)assigned in the loop at the end of loop execution
15.07 Construct a flowchart for nested structures using standard symbols and pseudocode	A decision indicating an outer structure (selection or repetition) is represented by a diamond filled with text as appropriate for the structure The outer decision has one input arrow from processes prior to the decision and two output arrows representing the Yes/No or True/False paths, and for a repetition structure one input arrow from processes inside the loop A decision indicating a nested structure (selection or repetition) is represented by a diamond filled with text as appropriate for the structure The nested decision has one input arrow from processes prior to the decision and two output arrows representing the Yes/No or True/False paths, and for a repetition structure one input arrow from processes inside the loop The Yes/No or True/False paths are clearly and appropriately labeled The arrows for each nested structure converge before returning flow to the immediate outer structure All variables used in the nested structures are initialized where appropriate Operations are included in the nested structures as required by the problem
15.08 Track a flowchart with nested structures	Describe the overall purpose of the nested structure in English Describe how the nested structure initiates and terminates in English Determine the correct number of iterations for each structure Determine the correct number of iterations for the overall nested structure Determine the value(s) of all variable(s) (re)assigned in the nested structure prior to the start of the nested structure (iteration 0) Determine the value(s) of all variable(s) (re)assigned in the nested structure at the end of each innermost structure iteration Determine the value(s) of all variable(s) (re)assigned in the nested structure at the end of the nested
15.09 Create test cases to evaluate a flowchart	Create a thorough set of test cases to test all possible paths in a flowchart Use English to completely describe each test and the intended path through the flowchart List the test values in an appropriate format Test values are consistent with the test description
15.10 Construct a flowchart using standard symbols and pseudocode	Recognize and implement standard flowchart symbols • Start and stop for the overall flowchart are represented by ovals • Inputs and outputs are represented by parallelograms • Decisions are represented by diamonds

	• Processes, such as calculations, are represented by rectangles • Operations are connected with arrows with points at one end to indicate flow Arrows must connect all flowchart elements and indicate a continuous flow from start to stop Arrows must converge prior to stop so that there is only one arrow into the stop Overall flowchart ends in one single stop Text within the symbols is in concise English (not code or only math) that conveys the purpose of the step Flowchart is complete and represents all possible outcomes required by the problem Decisions are accompanied by Yes/No or True/False text on the appropriate arrows

16.00 Create and troubleshoot a selection structure

Learning Objective	Evidence
16.01 Convert between these selection structure representations: English, a flowchart, and code	Recognize that a diamond structure with one input arrow and two output arrows (labeled Yes/No or True/False) translates to an if or elseif statement The number of diamonds in the flowchart translates exactly to the number if and elseif statements Recognize that the first 1-in/2-out diamond in a flowchart (or first following other non-decision instructions or the first on a Yes path following a decision) is an if statement Recognize that all immediately following 1-in/2-out diamonds on the No or False path are elseif statements Recognize an else statement is implied if there are operations between the only or last diamond and the convergence of the flowchart connecting lines. Recognize that a convergence of the entire No or False path with the entire Yes or True path translates to an end statement
16.02 Code a selection structure	Begin a selection structure with an if The if is accompanied by a condition for which a true result corresponds to code that immediately follows elseif is used for a series of related conditions Each elseif is accompanied by a condition which a true result corresponds to code that immediately follows elseif is a single word – there is no space between else and if An else is used to handle any condition(s) not addressed in the earlier parts of the selection structure and not used if no code is needed before the end An else is not accompanied by a condition end is used to terminate the selection structure Statements between the if, elseif, else, and end are indented A selection structure addresses all necessary paths for a given problem

Exam 3: Learning Objectives under Evaluation

16.03 Track execution of a single selection structure	Identify correct path given the input value(s) Provide the correct output(s) for the path
16.04 Track execution of a nested selection structure	Identify correct paths given the input value(s) Provide the correct output(s) for the paths
16.05 Create test cases to evaluate a selection structure	Create a thorough set of test cases to test all possible paths in the selection structure Use English to completely describe each test and the intended path through the selection structure List the test values in an appropriate format Test values are consistent with the test description

Exam 3: Learning Objectives under Evaluation

17.00 Create and troubleshoot a repetition structure

Learning Objective	Evidence
17.01 Identify when an indefinite versus a definite looping structure should be used	Recognition that an indefinite looping structure is used when a condition must be met to terminate repeating operations Recognition that a definite looping structure is used when the number or iterations of operations is known or can easily be predetermined or established with starting, increment, and ending values
17.02 Convert between these indefinite looping structure representations: English, a flowchart, and code	Recognize that a diamond structure with two input arrows (one from outside and one from inside the loop) and two output arrows (labeled Yes/No or True/False) translates to a while statement Recognize variables that must be initialized before the while loop for the while conditional statements and operations within the loop to execute Variables used in the condition and within the loop are initialized in the same way across representations (e.g., same values) Recognize that the Yes or True path leads to operations within the while loop Operations are completed and ordered in the same way across representations Variables are (re)assigned within the loop in the same way across representations (e.g., same computations) Recognize that the No or False path translates to an end statement
17.03 Code an indefinite looping structure	Begin an indefinite looping structure with a while The while is followed by a condition for which a true result corresponds to code within the indefinite looping structure Variables in the condition are set correctly prior to the indefinite looping structure Variables assigned in recursive calculations within the indefinite loop are initialized correctly outside of the indefinite looping structure Variables in the condition are updated in the indefinite loop in such a way as to make the condition false Operations within the indefinite looping structure are correct end is used to terminate the indefinite looping structure Statements between the while and end are indented
17.04 Track execution of an indefinite looping structure using a variable tracking table	Describe the overall purpose of the indefinite loop in English Describe how the indefinite loop initiates and terminates Determine the correct number of iterations Determine the value(s) of all variable(s) (re)assigned in the loop prior to the start of the loop (iteration 0) Determine the value(s) of all variable(s) (re)assigned in the loop at the end of each iteration Determine the value(s) of all variable(s) (re)assigned in the loop at the end of loop execution

Exam 3: Learning Objectives under Evaluation

17.05 Convert between these definite looping structure representations: English, a flowchart, and code	Recognize that a diamond structure with three entries and two input arrows (one coming from outside and one coming from inside the loop) and two output arrows (labeled Yes/No or True/False) translates to a for statement The three entries in the diamond correspond to the start value: increment : end value in the for statement Variables used within the loop are initialized in the same way across representations Recognize that the Yes or True path leads to operations within the for loop Variables are (re)assigned within the loop in the same way across representations Operations are completed and ordered in the same way across representations Recognize that the No or False path translates to an end statement
17.06 Code a definite looping structure	Begin a definite looping structure with a for Correct syntax for a for is index = start_value:increment:end_value or index = vector Variables assigned in recursive calculations in the definite loop are initialized correctly outside of the definite looping structure Operations within the definite looping structure are correct end is used to terminate the definite looping structure Statements between the for and end are indented
17.07 Track execution of a definite looping structure using a variable tracking table	Describe the overall purpose of the definite loop in English Describe how the definite loop initiates and terminates Determine the correct number of iterations Determine the value(s) of all variable(s) (re)assigned in the loop prior to the start of the loop (iteration 0) Determine the value(s) of all variable(s) (re)assigned in the loop at the end of each iteration Determine the value(s) of all variable(s) (re)assigned in the loop at the end of loop execution
17.08 Eliminate unnecessary definite looping structures	Recognize when a definite loop structure can be replaced with an element-by-element array operations Replacement of definitive loop structure with equivalent element-by-element array operations

Exam 3: Learning Objectives under Evaluation

18.00 Create and troubleshoot a repetition structure that employs vector indexing

Learning Objective	Evidence
18.01 Code an indefinite looping structure that employs vector indexing	Begin an indefinite looping structure with a while The while is followed by a condition for which a true result corresponds to code within the indefinite looping structure Variables in the condition are set correctly prior to the indefinite looping structure Variables assigned in recursive calculations within the indefinite loop are initialized correctly outside of the indefinite looping structure Vector index variables(s) are initialized outside of the indefinite looping structure Variables in the condition are updated in the indefinite loop in such a way as to make the condition false Vector index variables(s) are updated in the indefinite loop Operations within the indefinite looping structure not involving the vector index variables(s) are correct Operations within the indefinite looping structure involving vector index variables(s) (such as building or replacing values in a vector) are correct end is used to terminate the indefinite looping structure Statements between the while and end are indented
18.02 Track an indefinite looping structure that employs vector indexing using a variable tracking table	Describe the overall purpose of the indefinite loop that employs vector indexing in English Describe how the indefinite loop that employs vector indexing initiates and terminates in English Determine the correct number of iterations Determine the value(s) of all variable(s) (re)assigned in the indefinite loop prior to the start of the loop (iteration 0) including all values in vector variables Determine the value(s) of all variable(s) (re)assigned in the indefinite loop at the end of each iteration including all values in vector variables Determine the value(s) of all variable(s) (re)assigned in the indefinite loop at the end of loop execution including all values in vector variables
18.03 Code a definite looping structure that employs vector indexing	Begin a definite looping structure with a for Correct syntax for a for loop control statement is for index = start_value:increment:end_value or for index = vector Variables assigned in recursive calculations in the definite loop are initialized correctly outside of the definite looping structure Any vector index variable(s) **that are not part of the for loop control statement** is initialized outside of the definite looping structure Any vector index variable(s) **that are not part of the for loop control statement** is updated in the definite loop Operations within the definite looping structure not involving the vector index variables(s) are correct Operations within the definite looping structure involving vector index variable(s) (such as building or replacing values in a vector) are correct end is used to terminate the definite looping structure Statements between the for and end are indented

18.04 Track a definite looping structure that employs vector indexing using a variable tracking table	Describe the overall purpose of the definite loop that employs vector indexing in English Describe how the definite loop that employs vector indexing initiates and terminates in English Determine the correct number of iterations Determine the value(s) of all variable(s) (re)assigned in the definite loop prior to the start of the loop (iteration 0) including all values in vector variables Determine the value(s) of all variable(s) (re)assigned in the definite loop at the end of each iteration including all values in vector variables Determine the value(s) of all variable(s) (re)assigned in the definite loop at the end of loop execution including all values in vector variables

19.00 Create and troubleshoot nested repetition structures

Learning Objective	Evidence
19.01 Convert between these nested structures representations: English, a flowchart, and code	Correctly identify a diamond structure as a selection, indefinite looping, or definite looping structure Correctly identify the outer and inner structures Recognize variables that must be initialized before selection and indefinite structure conditional statements and operations within the nested structures Variables used in the condition and within the nested structures are initialized in the same way across representations (e.g., same values) Recognize that for indefinite and definite looping structures the Yes or True paths lead to operations within these structures Operations are completed and ordered in the same way across representations Variables are (re)assigned within the nested structure in the same way across representations (e.g., same computations) Recognize flow and symbol indicators that translate to end statements
19.02 Code nested structures	An appropriate outer structure is selected for the problem context An appropriate inner structure is selected for the problem context Variables are initialized as appropriate for successful execution of the nested structure Variables are updated as appropriate for successful execution of the nested structure Each structure is terminated with an end Indentation of the code clearly demarcates inner and outer structures
19.03 Track nested structures using a variable tracking table	Describe the overall purpose of the nested structure in English Describe how the nested structure initiates and terminates Determine the correct number of iterations for each structure Determine the correct number of iterations for the overall nested structure Determine the value(s) of all variable(s) (re)assigned in the nested structure prior to the start of the nested structure (iteration 0) Determine the value(s) of all variable(s) (re)assigned in the nested structure at the end of each innermost structure iteration Determine the value(s) of all variable(s) (re)assigned in the nested structure at the end of the nested structure execution

Exam 3: Learning Objectives under Evaluation

19.04 Code nested looping structures that employ array indexing	Variables are created to index the row and column of array(s) One repetition structure is designated to track the row index One repetition structure is designated to track the column index Array index variables are initialized as appropriate for successful execution of the nested structure Array index variables are updated as appropriate for successful execution of the nested structure Operations located within the nested structure involving the array index variables (such as building or replacing values in an array) are correct end is used to terminate both repetition structures Indention of the code clearly demarcates inner and outer repetition structures
19.05 Track a nested looping structure that employs array indexing using a variable tracking table	Describe the overall purpose of the nested looping structure that employs array indexing in English Describe how the nested looping structure that employs array indexing initiates and terminates Determine the correct number of iterations for the inner and outer repetition structure Determine the correct total number of iterations for the nested looping structure that employs array indexing Determine the value(s) of all variable(s) (re)assigned in nested looping structure that employs array indexing prior to the start of the loop (iteration 0) including all values in array variables Determine the value(s) of all variable(s) (re)assigned in the nested loop at the end of each innermost looping structure iteration including all values in array variables Determine the value(s) of all variable(s) (re)assigned in the nested loop at the end of loop execution including all values in array variables