MATLAB and VBA Basics for Examinations

Topic	MATLAB	VBA
Case sensitive (is a different from A?)	Yes	No, but editor will change your typing to standard upper and lower case notation for VBA
Statement structure	Normally one statement per line, use three periods(…) to continue a statement on a new line	Normally one statement per line, use space+underscore(_) to continue a statement to a new line
Multiple statements per line	End initial statement(s) with semicolon or comma	End initial statement(s) with colon
Command window results	Results of all statements without semicolon shown in command window	N/A
Common variable types	Double by default, string	Long, Boolean, Double, Date, String
Declaration statement	Not required	Dim <variable> As <type>
Symbolic Constants	N/A	Const PI As Double = 3.14159265358979
Common arith-metic operators	Addition(+), exponentiation(^), subtraction and unary minus(–), multiplication(*) and division(/)	
Other arithmetic operators	Use function Mod(a,b)	Integer division{\), Mod
Array arithmetic operators	+ – * / ^ .* ./ .^ Note difference between matrix operators (*, /, ^) and term-by-term operators (.*, ./, .^)	Requires For loops. Two loops required for term-by-term operations and three loops for matrix multiplication
Solving **Ax** = **b**	x = A \ b	mmult(minverse(A),b)
Declaring Arrays	N/A	Dim <array>(<dimensions>) as <type>, where dimensions can be a single number or a range like 5 To 12.
Lowest array subscript	1	0 by default, default can be changed to 1 by Option Base 1 statement and can be set to any value for individual arrays
Array components	A(i), B(i,,j)	
Entering Arrays	Row array: A = [1 2 3 …] Column array B = [1; 2; 3; …] Rectangular C = [1 2; 3 4; 5 6;…]	N/A
Subarrays	D = A(row1:row2,col1:,col2) D = A(:,col1:col2) D = A(row1:,row2,:) D = A(oneRow, col1:col2) D = A(row1:row2,oneColumn)	For row = row1 To row2 For col = col1 to col2 D(row–row1+1, col–col1 +1) = A(row,col) Next col Next row
Common relational operators	Less than(<), less than or equal(<=), greater than or equal(>=), greater than(>)	
Different relational operators	Equal (==), not equal(~=)	Equal(=), not equal(<>)
Scalar logical operators	Not(~), and(&&), or(||)	Not(Not), and(And), or(Or)
Array logical operators	Not(~), and(&), or(|)	N/A
Array Example	x = 0:100 y = sin(pi*x/100)	Dim x(0 To 100 As Double Dim y(0 To 100) As Double For k = 0 To 100 x(k) = k y(k) = sin(PI*k/100) Next k

MATLAB and VBA Basics for Examinations

Topic	MATLAB	VBA

If statements	if <condition1>
	  <Done if condition1 is true>
	elseif <condition2>
	  <Done if condition2 is true>
	elseif <condition3>
	  <Done if condition3 is true>
	<May be other conditions>
	else
	  <Done if all conditions false>
	end
	<Execute here after any statements done>
		If <condition1> Then
		  <Done if condition1 is true>
		ElseIf <condition2> Then
		  <Done if condition2 is true>
		ElseIf <condition3> Then
		  <Done if condition3 is true>
		<May be other conditions>
		Else
		  <Done if all conditions false>
		End If
		<Execute here after any statements done>

Count-controlled loop	for <counter> = <array> <statements> end	For <counter> = <start> To _ <end> Step <increment> <statements> Next <increment>
Similar count-controlled loop	for <counter> = <start>: .. <increment>:<end> <statements> end	Same for loop as above
Basic conditional loop	while <condition> <statements> end	Do While (<condition>) <statements> Loop
Other conditional loops	Can create with combinations of if statements (to allow test after) and while loop. Necessary to change condition that remains false in until loop to one that remains true in while loop/	Do <statements> Loop Do Whilel (<condition>) <sts> Loop Do Until (<condition>) <sts> Loop Do <sts> Loop Until (<condition>) Do <sts> Loop While (<condition>)
Functions	function <return> = <name> … (<arguments>) <return> is a one variable or row array of variables returned by the function <name> is the name of the function <arguments> may be blank or have one or more variable names separated by commas Arguments provide input data to function Each variable in the <return> list must be assigned a value in the function	Function <name> (<arguments>) As <type> <name> is the name of the function <type> is the data type for the function <arguments> may be blank or have one or more entries of the form: <variable> As <type> <variable> is a variable used in the function <type> is the data type for that variable Separate multiple <variable> entries in the <arguments> list by commas Arguments provide input data to function Set function name equal to return value
Strings	s = ‘this is a string’	Dim s as String : s = “This is a string”
Concatenation	s = [‘this is’ ‘ a string’]	s = ‘This is a ‘ & ‘string’ (can use + instead of &)