**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **6/July/2020** | **Name:** | **Prashantha naik** |
| **Course:** | **MATLAB Onramp** | **USN:** | **4al17ec074** |
| **Topic:** | **Course Overview,** **Commands,** **MATLAB Desktop and Editor,** **Vectors and Matrices** | **Semester & Section:** | **6th b** |
| **GitHub Repository:** | **prashanth\_course** |  |  |

|  |
| --- |
| **SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  Create Symbolic Numbers  You can create symbolic numbers by using sym. Symbolic numbers are exact representations, unlike floating-point numbers.  Create a symbolic number by using sym and compare it to the same floating-point number.  sym(1/3)  1/3  ans =  1/3  ans =  0.3333  The symbolic number is represented in exact rational form, while the floating-point number is a decimal approximation. The symbolic result is not indented, while the standard MATLAB® result is indented.  Calculations on symbolic numbers are exact. Demonstrate this exactness by finding sin(pi) symbolically and numerically. The symbolic result is exact, while the numeric result is an approximation.  sin(sym(pi))  sin(pi)  ans =  0  ans =  1.2246e-16  Vector, Matrix and Array Commands  The following table shows various commands used for working with arrays, matrices and vectors −   |  |  | | --- | --- | | **Command** | **Purpose** | | cat | Concatenates arrays. | | find | Finds indices of nonzero elements. | | length | Computes number of elements. | | linspace | Creates regularly spaced vector. | | logspace | Creates logarithmically spaced vector. | | max | Returns largest element. | | min | Returns smallest element. | | prod | Product of each column. | | reshape | Changes size. | | size | Computes array size. | | sort | Sorts each column. | | sum | Sums each column. | | eye | Creates an identity matrix. | | ones | Creates an array of ones. | | zeros | Creates an array of zeros. | | cross | Computes matrix cross products. | | dot | Computes matrix dot products. | | det | Computes determinant of an array. | | inv | Computes inverse of a matrix. | | pinv | Computes pseudoinverse of a matrix. | | rank | Computes rank of a matrix. | | rref | Computes reduced row echelon form. | | cell | Creates cell array. | | celldisp | Displays cell array. | | cellplot | Displays graphical representation of cell array. | | num2cell | Converts numeric array to cell array. | | deal | Matches input and output lists. | | iscell | Identifies cell array. |  What's an Array?   All MATLAB variables are *arrays*. This means that each variable can contain multiple elements. You can use arrays to store related data in one variable.    Because you'll use arrays every time you program, it's important to get to know them and the terminology used to describe them.    For long vectors, entering individual numbers is not practical. An alternative, shorthand method for creating evenly-spaced vectors is to use the : operator and specify only the start and end points.  y = 5:8  y =  5 6 7 8  MATLAB contains many functions that help you to create commonly used matrices, such as matrices of random numbers.  x = rand(2)  x =  0.8147 0.1270  0.9058 0.9134  Note that the 2 in the command rand(2) specifies that the output will be a 2-by-2 matrix of random numbers. |