**CSCI 2150L**

**Session 4 Topics:**

We started the session by generating a matrix. We further discussed the indexing pattern of a 2-D matrix in MATLAB using the following generic representation of a 2-D matrix.

[Note: Observe the indexing pattern for each row and column. For any column only the first index i.e. the row index changes; whereas for any row only the column index varies.]

1. To access any element of the matrix we used the command **Ar(row,col)**, where **Ar** is the name of the array, **row** and **col** being the row index and column index respectively. For e.g. to access the element a2m from the matrix above we have to use command Ar(2,m).
2. To find diagonal elements of a matrix (those elements whose row and column index are the same) we used **diag(Ar)** command.
3. We discussed transpose of a matrix where the row and column index interchange. To generate a transpose matrix in MATLAB we used command (note the small single quote above the matrix). Transpose of the above matrix would be as follows:

[Notice the change in relative arrangement of elements in Ar and Ar’]

1. We discussed briefly about a special square matrix, namely identity matrix whose all diagonal elements are ones and other elements are zeros. Following is an identity matrix.

Identity matrices often use the symbol .

1. Then we generated some square matrices with all zeros and all ones using commands **zeros(dim)** and **ones(dim)** command where **dim** is the dimension of the square matrix.
2. Matrices with only one row and only one column often referred to as row vector and column vector respectively. We generated two row vectors **x** and **y** using commands as follows:

X = 1:5;

Y = 6:10;

Such that x = [1 2 3 4 5] and y = [6 7 8 9 10] have only one row.

1. We generated some special matrices using x and y as follows
2. z1 = [x y]; b. z2 = [x; y]; c. z3 = [x’ y’]; d. z3 = [x’; y’]; e. z4 = [z3 y’]

[Note that if x and y are row matrices the their transpose x’ and y’ would be column matrices. Try to practice these commands to get familiarized with different ways to generate a matrix using other matrices.]

1. At last we extracted a sub array from a 2-D matrix using command **Ar(row, col\_min: col\_max)** where **Ar** is the name of the matrix, **row** is index of the row in which the sub array exists, **col\_min** and **col\_max** specifies the minimum and maximum index of the sub array to be extracted from Ar. For e.g. if we want to extract elements a23 to a2m from matrix **Ar** in 1 Then we would use the following command **Ar(2, 3:m)**.