Review of MATLAB® Commands and Syntax

Input Statements:

```
% Prompt the user to enter a number and store the number under the
variable name, x, as a double
x = input('Enter a number please:  ');

% Prompt the user to enter text and store the number under the
variable name, month, as a string
month = input('Enter the month that you were born: ','s');

% Create a pop-up menu with directions & three buttons: Yes,No,What?
% If the user clicks on the Yes button, then the variable choice = 1
% If the user clicks on the No button, then the variable choice = 2
% If the user clicks on the What? button, then the variable choice = 3
choice = menu('Do you enjoy programming','Yes','No','What?');
```

Output Statements:

```
% Decimal Numbers
x = 75.176;
fprintf('The variable x = %0.2f  \n',x);
% %0.2f means show two places behind decimal point

The variable x = 75.18

% Integers
y = 750;
fprintf('The value of y is:  %i \n',y);

The value of y is:  750

% Strings
course = 'Models II';
fprintf('The name of this course is:  %s \n',course);

The name of this course is:  Models II

% Mixture
x = 75.176; y = 750; course = 'Models II';
fprintf('x = %0.1f, y = %i, course is %s \n',x,y,course');

x = 75.2, y = 750, course is Models II
```

Conditional Statements:

```
Assume variables a and b are defined in MATLAB (as numbers)

% Numbers (Sample Conditions for numbers)
a == 10;             % True if a is equivalent to 10
a >= 10;             % True if a is ≥ 10
a >= 2 && a <= 15;   % True if 2 ≤ a ≤ 15
a >= 2 && b == 3;    % True if a ≥ 2 and b is equivalent to 3
a >= 2 || b == 3;    % True if a ≥ 2 or b is equivalent to 3

% Strings (Sample Conditions for strings)
month = 'July';
strcmp(month,'July');    % True
strcmp(month,'August'); % False
strcmp(month,'july');    % False
strcmpi(month,'july');   % True
```

if … elseif … elseif … else:

```
if Condition1
    % MATLAB Commands to execute if Condition1 is TRUE
elseif Condition2
    % MATLAB Commands to execute if Condition1 is FALSE
    % and Condition2 is TRUE
elseif Condition3
    % MATLAB Commands to execute if Condition1 is FALSE
    % and Condition2 is FALSE
    % and Condition3 is TRUE
else
    % MATLAB Commands to execute if Conditions 1-3 are FALSE
end
```

Switch Statements:

```
switch variable    % variable must be defined (number or string)
    case value1
        % MATLAB Commands to execute if variable is value1
    case value2
        % MATLAB Commands to execute if variable is value2
    case value3
        % MATLAB Commands to execute if variable is value3
    otherwise
       % MATLAB Commands to execute if variable is none of above
end
```

For Loops:

```
% Assume variable N is defined (an integer)
for k = 1:N

    % MATLAB Commands to execute a total of N times
    % Counter index, k, keeps track of number of times thru loop

end
```

While Loops:

```
while Condition

    % MATLAB Commands will execute again and again as long as
    % Condition is true

end
```

Arrays:

Suppose x is a vector (1-d Array)

```
x(3) is the 3rd entry in vector, x

x(3) = 5        % Puts a 5 in the 3rd entry of vector x

N = length(x)   % N = the number of entries in vector x
```

Suppose M is a matrix (2-d Array)

```
M(3,4) is the entry in the 3rd row and 4th column of M

M(2,3) = 5                % Puts a 5 in 2nd row and 3rd column of M

[Rows Cols] = size(M)     % Rows = number of rows in M
                          % Cols = number of columns in M
```

Summary of Some Useful Array Functions

Assume x is a vector (1-d array), M is a matrix (2-d arry)

Function	Description
zeros(1,N)	Creates a vector of length N filled with zeros
min(x)	Smallest entry in vector, x
max(x)	Largest entry in vector, x
length(x)	Number of entries in vector, x
sum(x)	Add all values in vector, x
find(*ConditionOn_x*)	finds all entry numbers in x where condition is true For example, find(x > 3) would find the location of all entries in vector x that are greater than 3
zeros(r,c)	Creates a matrix with r rows and c columns filled with zeros
min(M)	Smallest entry in each column of matrix M
min(M,[ ],2)	Smallest entry in each row of matrix M
min(min(M))	Smallest entry in the matrix M
max(M)	Largest entry in each column of matrix M
max(M,[ ],2)	Largest entry in each row of matrix M
max(max(M))	Largest entry in the matrix M
size(M)	Number of rows and Number of columns in matrix M
sum(M)	Sums each column of matrix M
sum(M,2)	Sums each row of matrix M
sum(sum(M))	Sums all of the entries in matrix M