Flow Control & Logic in Matlab

See Chapter 4 in your textbook.
Basic Concepts

These basic elements can be combined to create complex program logic.
The “if” Statement

**Basic syntax:**

```matlab
if ( condition1 )
    % do some work
elseif ( condition2 )
    % do different work
    ;
else
    % do default work
end
```

```matlab
if ( condition )
    % do some work
end
```

Create a MATLAB script to plot cos(x) and sin(x) on a user-specified interval. The user should be able to enter the interval in degrees or radians.
Relational Operators

- **True** condition represented by a nonzero (typically “1”).
- **False** condition represented by zero “0”
- Can be applied to scalars, vectors, or matrices.

### Comparison Operators

<table>
<thead>
<tr>
<th>Statement</th>
<th>Result</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>a == b</td>
<td>true if a and b are equal</td>
<td>5==3 false</td>
</tr>
<tr>
<td>a ~= b</td>
<td>true if a and b are NOT equal</td>
<td>5~=3 true</td>
</tr>
<tr>
<td>a &lt; b</td>
<td>true if a is less than b</td>
<td>5&lt;3 false</td>
</tr>
<tr>
<td>a &gt; b</td>
<td>true if a is greater than b</td>
<td>5&gt;3 true</td>
</tr>
<tr>
<td>a &gt;= b</td>
<td>true if a is not less than b</td>
<td>5&gt;=3 true</td>
</tr>
<tr>
<td>a &lt;= b</td>
<td>true if a is not greater than b</td>
<td>5&lt;=3 false</td>
</tr>
</tbody>
</table>

### Logical Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>Element-wise AND - returns an array of 1 and 0.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>~</td>
<td>Element-wise NOT - returns an array of 1 and 0</td>
</tr>
</tbody>
</table>

Tuesday, September 23, 2008
Example: What does this do?

dice = 3*rand(1); % a number between 0 and 3
if ( dice<1 )
    name = 'Bob';
elseif (dice<2)
    name = 'Fred';
else
    name = 'Jane';
end

dice = 3*rand(1); % a number between 0 and 3
if dice<1
    age = 25;
elseif dice<2
    age=19;
else
    age = 40;
end

fprintf('
%s is %1.0f years old\n\n',name,age);
## A Few More Useful Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>any(var)</td>
<td>returns true if any element of var is true</td>
</tr>
<tr>
<td>all(var)</td>
<td>returns true (1) if all elements of var are true</td>
</tr>
<tr>
<td>find(var)</td>
<td>returns the indices where var is true (nonzero)</td>
</tr>
<tr>
<td>isequal(var1, var2)</td>
<td>returns true (1) if the two arrays are equal.</td>
</tr>
<tr>
<td>strcmp(str1,str2)</td>
<td>Compares two strings and returns true if they are equal.</td>
</tr>
<tr>
<td>abs(var)</td>
<td>returns the absolute value of all elements of var.</td>
</tr>
<tr>
<td>ceil(var)</td>
<td>rounds all elements of var up.</td>
</tr>
<tr>
<td>floor(var)</td>
<td>rounds all elements of var down.</td>
</tr>
<tr>
<td>mod(var1,var2)</td>
<td>Remainder of division of var1 by var2.</td>
</tr>
</tbody>
</table>
Example - Data Analysis

Generate a set of random numbers between 1 and 100.

- What percentage of these numbers are between 40 and 60?
- How many numbers did it take to get a consistent answer?

Repeat this example to determine what percentage are between 90 and 95.

Hint: use the `rand` function.
The “for” Statement

Predetermined looping

Basic syntax:

\[
\text{for(} \text{counter=} \text{start}: \text{step}: \text{stop}) \\
\quad \% \text{ do some work} \\
\text{end}
\]

Example - what are the values in \(a\)?
\[
\begin{align*}
n &= 5; \\
a &= \text{zeros}(n,1); \\
\text{for } i &= 1:n \\
& \quad a(i) = 2*i; \\
\text{end}
\end{align*}
\]
The “while” Statement

Conditional Looping

**Basic syntax:**

```c
while ( condition )
    % do some work - must result
    % in condition being changed
    % at some point!
end
```

**Example - What is the value of n?**
```
a = 1;
n = 0;
while (a<10 )
    a = a+2;
    n = n+1;
end
```
Example: Factorial

\[ n! = \prod_{i=1}^{n} i \]

Write a Matlab code to calculate the factorial of a number using:

1. A `for` loop
2. A `while` loop

**NOTE:** MATLAB’s factorial function will do this much faster than using loops will.
Define two vectors. Have the user choose one of two options:

1. Calculate the dot product of two vectors using loops

2. Calculate the elemental product of two vectors using loops.
Example: Craps

Given a “bet,” determine how many rolls of the dice you must have to win.

- Two dice: what bets are allowable? (Prevent invalid bets)
- How would we set this up?