Lesson 6: Decision Structure (Branching)

Topics:

- if-then-else statements
- Relational expressions
- Logical Statements
- Compound Statements

Learning Outcomes:

1. Represent decisions programmatically

Book Sections: 4.1 - 4.3

1. Logical Statements

Logical statements, (a.k.a. Boolean Expressions, conditional statements) are frequently used in programming for decision making. They control the flow of the program in order to execute certain blocks of code depending on the truth value of the expression.

1.1 Relational Operators

Recall the relational operators previously introduced:

```
% <         Less than
% >         Greater than
% <=        Less than or equal
% >=        Greater than or equal
% ==        Equal
% ~=        Not Equal
```

Here are some logical statements that evaluate to TRUE or FALSE:

```
1<2             % True
isprime(4)      % False
4==3            % False
isinteger(5)    % False (Why?)
```

```
rem(13,4)==2    % False
```

2. Compound Statements

Logical statements can be compounded using AND and OR statements.

```
% Syntax for AND:  &&
% Syntax for OR:   ||

% Example: AND
1<2 && isinteger(5)     % False

% Example:  OR
1<2 || isinteger(5)     % True

% Example: AND and OR
1<2 && (3<4 || 4==3)

% Question: Do parenthesis matter?
```

3. if-then Statements

In MATLAB, you actually do not type "then". However, you DO need an "end" statement with each if statement.

```
% Basic if-then Statement
if condition
    % run this
end

% This says, if the "condition" is TRUE, then
% run the code inside (indented).

% Example:
x=sin(3);
if x<0.3
    disp('Statement is true');
end
```

3.1 Compound Statement

```
x=sin(5);
y=cos(5);

% AND
if x>0 && y<0
    disp('Both conditions are true')
end


% OR
if x>0 || y<0
    disp('Either condition is true')
end
```

4. if-then-else

Many times you will need to explicitly state what will happen when then condition is false. In these cases, use the "if-then-else" statement.

```
% Determine if a function value is positive or not

x=sin(5);

if x>0
    disp('x is positive');
else
    disp('x is NOT positive');
end
```

4.1 if-elseif

When there are multiple conditions to check use "elseif".

```
% Determine if a value is positive, negative, or zero

x=sin(5);
```

```
x = 5;

if x<0
    disp('x is less than zero');
elseif x==0
    disp('x is equal to zero');
elseif x>0
    disp('x is greater than zero');
end
```

5. Examples

```
% Determine if a number is divisible by 4.
% For example, 4, 8, 12, 16, 20, 24, ...
n=input('Enter a number');

if rem(n,4)==0
    disp('Your number is a multiple of 4');
end
```

```
% Determine the number of solutions to a quadratic equation
% a*x^2+b*x+c=0.

% Get the coefficients
a=1;
b=-5;
c=6;

% Calculate the determinant
D=b^2-4*a*c;

% Decide number of solutions
if D<0
    disp('0 Solutions');
elseif D==0
    disp('1 Solution');
elseif D>0
    disp('2 Solutions');
```

```
end
```

```
% Calculate the area of a circle

r=input('Enter the radius');

% Check if radius is a positive number
if r>0
    A=pi*r^2;
else
    disp('Radius is invalid');
end
```

Return to MAT225