# Functions

Mili I. Shah

# Relational and Logical Operators

- Relational operator compares two numbers and states either true (1) or false (0)

| | |
|---|---|
| < | Less than |
| > | Greater than |
| <= | Less than or equal |
| <= | Greater than or equal |
| == | Equal to |
| ∼= | Not Equal to |

- Logical operator examines true/false statements and produces true (1) or false (0)

| | | |
|---|---|---|
| & | AND | Both true result is true (1) otherwise (0) |
| \| | OR | Either or both are true then true (1). If both false then false (0) |
| ∼ | NOT | Gives the opposite of the opperand. If operation false then true (1) otherwise false (0) |

# Conditional Statements

- A conditional statement is a command that allows Matlab to make a decision of whether to execute a groups or commands

```
if (cond state)

_____Group 1 Commands

elseif (cond state)

_____Group 2 Commands

else

_____Group 3 Commands

end
```

# Conditional Statements

**Example**: A worker is paid according to his hourly wage up to 40 hours and 50% more for overtime. In addition, if the worker works more than 60 hours then the worker gets a $100 bonus.

```
t = input('Enter the number of hours worked ');
h = input('Enter the hourly wage ');
if t>60
   Pay = t*h;
   Pay = Pay + (t-40)*.5*h + 100;
elseif t>40
   Pay = t*h;
   Pay = Pay + (t-40)*.5*h;
else
   Pay = t*h;
end
fprintf('Your pay is $%5.2f\n',Pay);
```

# Conditional Statements

**You Try**: Ask a user to submit a number. Check if it is an integer. If the integer is even, print ‘This number is even.’ If not print ‘This number is odd’

[redacted]

Hint: consider: `floor` or `ceil`

# Conditional Statements

**You Try**: Ask a user to submit a number. Check if it is an integer. If the integer is even, print ‘This number is even.’ If not print ‘This number is odd’

```
num = input('Enter an integer: ');
if floor(num)-num ~= 0
    disp('Not an integer')
elseif mod(num,2) == 0
    disp('Integer is Even')
else
    disp('Integer is Odd')
end
```

# For-end Loops

In `for-end` loops the execution is repeated a number of times

```
for k = f:s:t
....
end
```

- f: first number
- s: increment by
- t = end number

```
>>for k = 1:3:7
k∧2
end
ans =
1
ans =
16
ans =
49
```

# For-end Loops

**You try:** Sum the integers from 1 to 15 incremented by 2.

[redacted]

# For-end Loops

**You try:** Sum the integers from 1 to 15 incremented by 2.

```
sumInt = 0;
for k = 1:2:15
  sumInt = sumInt + k;
end
sumInt = 64
```

You could also do `sum([1:2:15])`

# While-end Loops

In `while-end` loops the execution is repeated until a condition is satisfied

```
while (conditional statement)
....
end
```

```
>>k=1;
>>while k<7
k∧2
k = k+2;
end
ans =
1
ans =
9
ans =
25
```

# While-end Loops

**You Try:** Continue to look at fractions $\frac{1}{n}$ and stop when $\frac{1}{n} - \frac{1}{n+1} \leq 10^{-6}$

```
n = 1;
err = 1;
while err>1e-6
  n = n+1;
  err = 1/n - 1/(n+1);
end
```

# While-end Loops

**You Try:** Continue to look at fractions $\frac{1}{n}$ and stop when $\frac{1}{n} - \frac{1}{n+1} \leq 10^{-6}$

```
n = 1;
err = ones(10000,1);
while err(n)>1e-6
  n = n+1;
  err(n) = 1/n - 1/(n+1);
end
semilogy(err(1:n))
```

# Switch

`Switch` among several cases based on expression

```
switch switchExpression
_____case caseExpression
__________statements
_____case caseExpression
__________statements
....
_____otherwise
__________statements
end
```

```
mynumber = input(’Enter a number:’);
switch mynumber
case -1
 disp(’negative one’);

case 0
 disp(’zero’);

case 1
 disp(’positive one’);

otherwise
 t disp(’other value’);

end
```

# Notes on Nesting

- Loops and conditional statements can be nested within themselves
- Keep organized by having nice spacing (USE TABS)

```
nest1
_____nest2
__________nest3
__________end
_____end
end
```

# Initializing Data

- You could append data to a matrix. COSTLY $\ddot\frown$
- Instead initialize data and change

**Example**

- mat = [1 1 1 1]; mat = [mat;1 2 3 4]; $\ddot\frown$
- mat = ones(2,4); mat(2,:) = [1 2 3 4]; $\ddot\smile$

# Taylor's Series

Taylor's series of a real or complex function $f(x)$ that is infinitely differentiable at a real or complex number $a$ is the power series

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n \cdots$$

For example, $\sin(x)$ when a=0 can be represented as

$$\begin{aligned} \sin(x) &= \sin(0) + \cos(0)(x) - \frac{\sin(0)}{2!}(x)^2 \cdots + \frac{f^{(n)}(0)}{n!}(x)^n \cdots \\ &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \cdots \end{aligned}$$

# Taylor's Series

Taylor's series of a real or complex function $f(x)$ that is infinitely differentiable at a real or complex number $a$ is the power series

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n \cdots$$

For example, $\sin(x)$ when $a = \pi$ can be represented as

$$\begin{aligned} \sin(x) &= \sin(\pi) + \cos(\pi)(x) - \frac{\sin(\pi)}{2!}(x)^2 \cdots + \frac{f^{(n)}(\pi)}{n!}(x)^n \cdots \\ &= -x + \frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} \cdots \end{aligned}$$