

ECE 257 - LESSON 15

INTRODUCTION TO BRANCHING

SPRING 2007

A.P. FELZER

IN CLASS

IF STATEMENTS

1. If-else statements

```
x = input ('Enter a number x: ');
if x < 0
    disp ('The number x you entered is negative');
else
    disp ('The number x you entered is not negative');
end
```

- What happens when the if statement is satisfied
- What happens when the if statement is not satisfied
- What happens when the user inputs the character 'a'

2. Nested if statements

```
x = input ('Enter a number x: ');
if isnumeric (x)
    if x < 0
        disp ('The number x you entered is negative');
    else
        disp ('The number x you entered is not negative');
    end
else
    disp ('You did not enter a number');
end
```

- What are nested if statements and how do they work
- How many end's are needed

3. Elseif statements

```
x = input ('Enter a number x: ');
if x < 0
    disp ('The number x you entered is negative');
elseif x == 0
    disp ('The number x you entered is zero');
else
    disp ('The number x you entered is positive');
end
```

- How is using elseif different from using nested if's
- How many end's are needed
- Can elseif's be used if the options are not mutually exclusive

4. Replacing elseif's with separate if's

```
x = input ('Enter a number x: ');
if x < 0
    disp ('The number x you entered is negative');
end
```

```

if x == 0
    disp ('The number you entered is zero');
end
if x > 0
    disp ('The number x you entered is positive');
end

```

- a. What are the advantages and disadvantages of using separate if's rather than elseif's

SWITCH CONSTRUCT

5. Switch construct syntax

```

a = input ('Enter an integer between 1 and 6: ');
switch (a)
    case {1, 3, 5}
        disp ('You entered an odd integer');
    case {2, 4, 6}
        disp ('You entered an even integer');
    otherwise
        disp('You did not enter an integer between 1 and 6');
end

```

- a. How does the switch function work
- b. What's the purpose of 'otherwise'

TRY/CATCH

6. The try/catch construct for "trapping" errors so a program can continue without crashing

```

x = [1 -4 7];
try
    k = input ('Enter the subscript k of the element of vector x you want to display: ');
    x(k)
catch
    % If the program gets here an error has occurred
    disp ('The value of k you entered is not a valid subscript of the vector x')
end

```

- a. Describe how try/catch works
- b. Why is try/catch useful in writing user friendly software