Wichita State University
College of Engineering
Biomedical Engineering

Nested Loops and the Break Statement

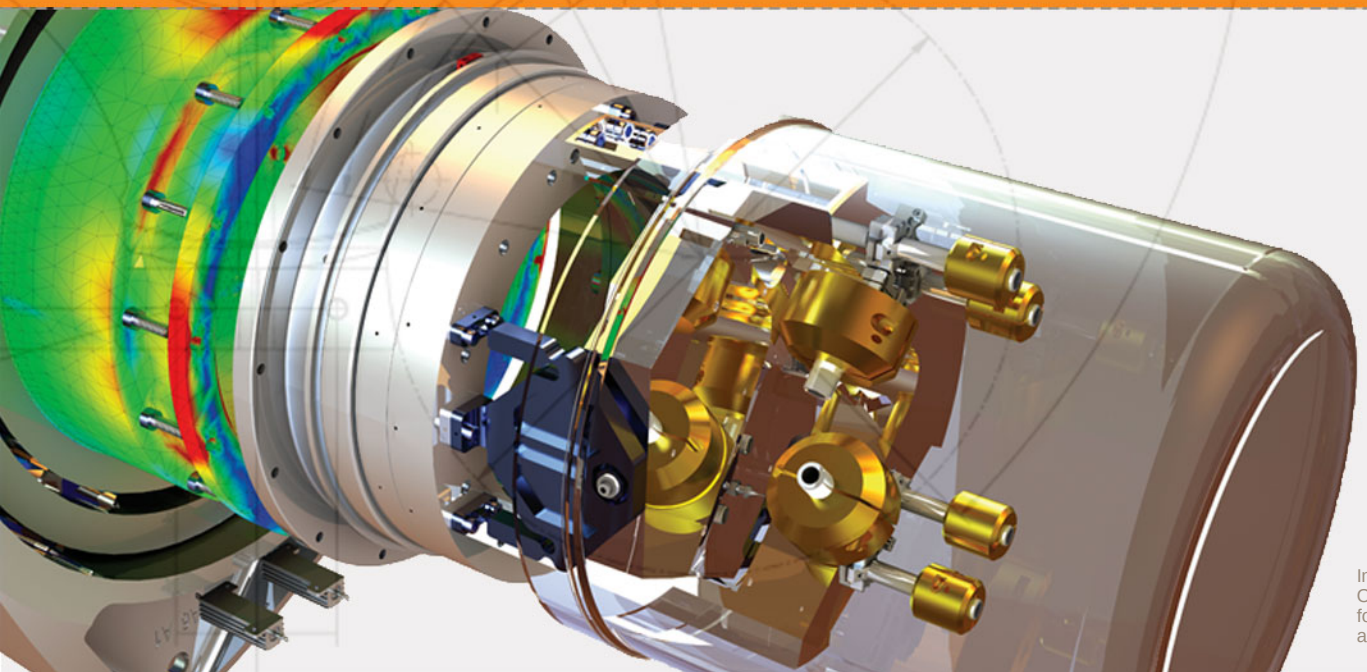

Image courtesy of National Optical Astronomy Observatory, operated by the Association of Universities for Research in Astronomy, under cooperative agreement with the National Science Foundation.

What are Nested Loops?

- Nested loops are:
  - Loops which run inside another loop

- When would you use nested loops?
  - Performing the same set of operations on different data sets
  - Performing a set of computations with multiple independent variables
  - Working with 2-dimensional arrays

Nested For Loops

Nested for loops consist of one (or more) for loops inside of another:

```
for m = start:inc:end
      MATLAB Statements;
      for n = start2:inc2:end2
         MATLAB Statements;
      end
      MATLAB Statements;
end
```

- When nesting for loops:
  - Loop control variables must have different names
  - Inner loop will execute completely before the next iteration of the outer loop

Example: Multiplication Tables

```
for m = 1:1:5
      for n = 1:1:5
         fprintf('%i x %i = %i\t',m,n,m*n);
      end
      fprintf('\n');
end
```

Nested While Loops

Nested while loops consist of one (or more) while loops inside of another:

```
while condition1
      MATLAB Statements;
      while condition2
         MATLAB Statements;
      end
      MATLAB Statements;
end
```

- When nesting while loops:
  - Loop control variables can be the same or different
  - Conditions can be the same or different

Example: Guessing Game

```
repeat = 1;
while repeat == 1
    my_numb = randi(10,1);
    guess = 0;
    while guess ~= my_numb
        guess = input('Guess my number: ');
        if guess ~= my_numb
            fprintf('Try again.\n');
        end
    end
    fprintf('You found my number!\n');
    repeat = menu('Play again?','Yes','No');
end
```

Mix and Match

- It is possible to nest for loops inside of while loops and vice versa, depending on your application
  - If you do this, you must pay attention to which loop will execute at which time and keep track of your loop control variables

```
while more lines in file
    get next line from file
    for n = 1:number of characters in line
        count the number of a's
    end
    display the number of a's
end
```

Contingently Nested Loops

- It is also possible to have nested loops which depend on the outer loop to determine the number of times the loop will iterate

```
for m = 1:5
    for n = 1:m
        fprintf('*');
    end
    fprintf('\n');
end
```

Output:

```
*
**
***
****
*****
```

Break Statement

Another option for exiting a for loop or a while loop is to use a break statement.

break forces MATLAB to terminate the current loop

```
sum = 1;
for k = 1:20
      sum = sum + 1/2^k;
      if (2 – sum) < 1e-5
            disp(k);
            break;
      end;
end;
```

When k = 17, (2-sum) < 0.00001 so loop terminates.

Break Statement

Note about the Break statement:

- If there is a loop inside a loop (nested loop), the break statement will only affect the loop in which the break statement occurs

```
for x = 1:5
    fprintf('%d\n',x);
    for y = 1:5
        if (x*y > 9)
            break;
        else
            fprintf('%d',y);
        end
    end
    fprintf('\n');
end
```

```
for x = 1:5
    fprintf('%d\n',x);
    for y = 1:5
        fprintf('%d',y);
    end
    fprintf('\n');
    if (x*y > 9)
        break;
    end
end
```