MATH 365
Loops and Conditional Statements

1 Introduction

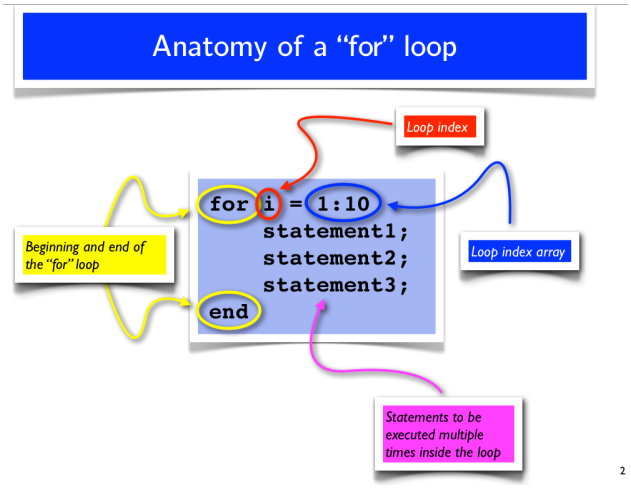
Every procedural programming language has conditional statements (if-statements) and iterative statements (loops). You should be familiar with the concepts so here we will only review them in order to understand how to control order of execution.

1.1 Loops

The **for loop** is a type of flow control statement that allows us to execute a set of commands multiple times. For example, in the following code fragment the statement *Hello World* gets executed 10 times, even though it only appears once in the code fragment.

```
for i = 1:10
    print("Hello, World!")
end
```

The for loop has three basic components: (i) the enclosing keywords *for* and *end* which enclose the set of statements to be carried out multiple times by the loop (the statement print("Hello, World!"), in the above example), (ii) a loop index which is the variable that stores the current value of the loop counter (i in the above example), and (iii) an index array containing the range of values over which the loop index should vary (the array 1:10 in the above example). See below for the Anatomy of a "for" loop.



Here's another example of a for loop

```
for count = 10:-5:1,
    print(count)
end
```

and it will print *10 5*.

The **while loop** is used to repeat a segment of code an unknown number of times until a specific condition is met. For example, say we want to know how many times a given number can be divided by 2 before it is less than or equal to 1. If we know a specific number, such as 32, we can say 5 times, but for a given symbolic variable NUMBER which represents any number in the world, how many times is not known a priori (before hand). In this case, we could use a while loop to determine that answer:

```
count = 0
while NUMBER > 1
    NUMBER = NUMBER /2
    count = count + 1
end
```

WARNING: If the action inside the loop does not modify the variables being tested in the loops condition, the loop will "run" forever. For example

```
while y < 10
    x = x + 1
end
```

is an infinite loop.

The **break statement** terminates execution of **for or while loops**. Statements in the loop that appear after the break statement are not executed. For example

```
fruits = ["apple", "banana", "cherry"]
for x in fruits
    if x == "banana"
       break
    end
    print(x)
end
```

will print *apple*. Pretend you are computer to see why this is the result!

The **continue statement** is used for passing control to the next iteration of a for or while loop. For example

```
fruits = ["apple", "banana", "cherry"]
for x in fruits
    if x == "banana"
       continue
    end
    print(x)
end
```

will print *apple cherry*. Again, pretend you are a computer so that you understand the logic of the conditional and iterative statements.

2 Individual Lab

Please answer the following questions in Blackboard.

1. How many times does the following loop print the statement *Three brown bears*?

```
for i = 1:10
    print("Three brown bears")
end
```

2. What is printed with the following statements?

```
fruits = ["apple", "banana", "cherry"]
for x in fruits
    print(x)
    if x == "banana"
       break
    end
end
```

3. What will print with the following statements?

```
for j = 2:3:5
    print(j)
end
```

3 Group work

Please complete the following questions in groups of 3-4 people. At the end of class, hand in your team's findings with each team member's name clearly written at the top of the page. Each team will consist of:

- The **manager**, responsible for coordinating the work of the team. The manager is the person who sitting furthest from the door.
- The **spokesperson**, who will report your group's work to the rest of the class. The spokesperson is the person who ate the healthiest breakfast.
- The **scribe**, who will be responsible for writing down your team's findings. The scribe is the person who is taking the fewest credit hours this semester.
- The **timekeeper**, who will keep track of time for each exercise. The timekeeper is the person who was not assigned a role already. In case of a three-person group, the manager takes the role of a timekeeper.

One person should not have more than one role (except for the manager/timekeeper combo), so if a person who already was assigned a task meets the criteria for another one, the person who is next in line (i.e., second-longest commute) takes this role.

1. (5 min)

   (a) Consider the pseudocode

   ```
   x = 4
   s = 0
   for i = 1:3
       s = s + x
   end
   ```

   What is the resulting value of s?

   (b) Consider the following pseudocode:

   ```
   x = 4
   s = 0
   for i = 1:n
       s = s + x
   end
   ```

   i. If you programmed it, there would be an error message. Why?

   ii. Since you cannot successfully program it, write down the sequence of values of s defined by the for loop. Once you identify a pattern in the sequence, write down a mathematical expression for s.

   The spokesperson should be prepared to report your answers written by the scribe to the rest of the class.

2. (5 min) Consider the following pseudocode:

   ```
   x = [0.2  0.15  0.004  0.55  0.77]
   findx = 100
   for i = 1:5
       if x(i) < findx
           findx = x(i)
       end
   end
   ```

   (a) What is the resulting value of findx?

   (b) How could you alter this code to find the maximum value in the array (vector) x?

   The spokesperson should be prepared to report your answers written by the scribe to the rest of the class.

3. (10 min) Consider the following pseudocodes:

   (a)

   ```
   yfor = 1
   for i = 1:n
       yfor = yfor*x(i)
   end
   ```

i. Identify two reasons why coding this would result in an error message.

ii. Even though we can't code it, write the sequence defined by the for loop by hand. Identify the pattern in this sequence and write down the mathematical expression for the sequence using $\prod$ notation.

(b)

```
s = 0
for i = 1:n
    s = s + x(i)
end
mys = s/n
```

i. Again we cannot successfully code this, but write the sequence defined by the for loop by hand and identify the mathematical expression for the sequence using Σ notation.

ii. Identify what mys represents.

(c)

```
mu = 0
for i = 1:n
    mu = mu + x(i)
end
mu = mu/n
s=0
for i = 1:n
    s = s + (x(i)-mu)^2
end
s = sqrt(s/(n-1))
```

i. Again we cannot successfully code this, but write the sequence defined by the second for loop by hand. Identify the mathematical expression for this sequence using Σ notation.

ii. Identify what s represents.

The spokesperson should be prepared to report your answers written by the scribe to the rest of the class.

4. (10 min) This pseudocode is a bit more challenging.

```
for j = 1:n
    s = 1
    for i = 1:j
        s = s*x(i)
    end
    yarray(j) = s
end
```

(a) Identify the reasons why we would get error messages.

(b) Notice that yarray if formed with two for loops and is an array, or vector. Consider only one iteration of the outer loop j, i.e. j=1, and write the sequence defined by the inner for loop by hand. Identify the mathematical expression for yarray(1).

(c) Now consider the second iteration in the outer loop, i.e. j=2, and identify the mathematical expression for yarray(2).

(d) Now consider the third iteration in the outer loop, i.e. j=3, and identify the mathematical expression for yarray(3) using $\prod$ notation.

(e) Identify the mathematical expression of the sequence formed by the inner for loop using $\prod$ notation for any value of j.

(f) Explain how your answer to 4e is different than your answer to 3(a)ii

The spokesperson should be prepared to report your answers written by the scribe to the rest of the class.