Loops/Iteration

- Used to repeat an action
- Must have a STOP condition
- Three flavors - for, while, do/while

Which loop to use?

- task with a specific number of repetitions
  - a for loop
- task with a indeterminate number of repetitions
  - use a while or a do/while loop

while loop review

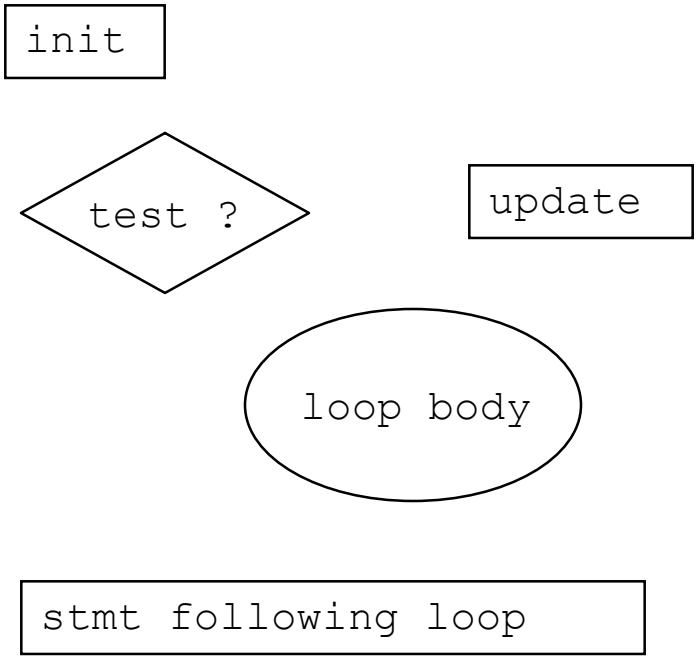
when is the test checked - after the body has finished executing or is it checked continuously ?

```
int n = 9;
while (n >= 0)
{
  n -= 3;
  System.out.println("A: "  + n );
  n -= 2;
  System.out.println("B: "  + n );
}
```

reminder -- The test is a "keep going" condition. Enter or do the loop again when the test is true.

Anatomy of a for loop

- 1 exec the init
- 2 exec the test
- 3 when the test is true
  - exec the body
  - exec the update
  - go back to step 2
- 3 when the test is false
  - exit the loop




```
int cnt;
for (cnt = 5; cnt <= 13; cnt += 3)
{
    System.out.print("*");
}
System.out.println("done");
```

for loop practice
what's the output?

```
for (int n = 13; n < 19; n += 2)
{
    System.out.print("*");
}
```

```
for (int a = 13; a >= 19; a += 2)
{
    System.out.print("*");
}
```

```
for (int n = 32; n >= 19; n -= 3)
{
    System.out.print("*");
}
```

for loop practice

```
for (int k = 12; k > 4; k -= 3 )
{
  if (k % 2 == 0)
        System.out.print(“foo“);
  else
        System.out.print(“bar“);
}
```

Nested for loops

```
ALWAYS use different for loop variables (i & j  in this example) for
   nested loops
int i,j;
for ( i=0; i <= 5; ++i )
{
   for (j=0; j <= 3; ++j )
   {
         System.out.print( “i: ”  + i + “,  j: “ + j“ );
   }
   System.out.println();
}
System.out.print(“After loops i= “ + i + “, j= “ + j );
```

// Why did we declare i and j outside (before) the loops?

A more complicated nested loop example

```
int a,b;
for (a = 50; a <= 54; ++a )
{
   System.out.println(“Start\n“);
   for (b = a + 10; b <= 63; b ++ )
   {
         System.out.println( a + “ “ + b);
   }
    System.out.println(“end”);
}
System.out.println(“After loops a= “ + a + “,  b= “ + b );
```

How to write your own loop

Sample problem statement: add numbers obtained from the user
until the sum of the numbers exceeds 1000 dollars.
Count the number of inputs (numbers) provided by the user.

- determine what variables will be needed - you may not think of them all at first, but this is a good place to start
- think of names for your variables. The more descriptive the names are the easier your program will be to understand.
  - a variable to hold the running total (sum)
  - a variable to count the number of inputs (count)
  - a variable to read the user's numbers into (number)

How to write your own loop (con't)

- decide when the loop will end
  - terminate when the sum > 1000

- negate the terminating condition to form the keep-going condition needed by your loop
  - sum <= 1000

- determine the kind of loop to use
  - use for loop for counting operations
  - use while loop for indeterminate situations

- set up your loop, plugging in the keep-going condition

```
while (sum <= 1000)
{

}
```

How to write your own loop (con't)

what needs to happen before the loops starts?

- initialize sum to 0
- initialize count to 0
- initialize number?

what goes in the loop?

what needs to be done multiple times?

- prompt for and read a number from user
- add number to sum
- increment counter

what should happen after the loop?

- report the info

```
int count = 0;
sum = 0;
while (sum <= 1000)
{
   System.out.print("Number please: ");
   int n= kbd.nextInt();
   sum += n;
   ++count;
}
System.out.println("sum:" + sum );
System.out.println("User entered " +
     count + " numbers ");
```