Topics: Iteration (while and for loops)

Reading (JV): Sec 3.4-3.6, 3.8

### The while loop

```
while ( condition )
    statement ;
```

### The for loop

```
for ( initialization; condition; increment )
    statement ;
```

### Pattern for doing something $n$ times

```
i = 1;
while ( i<=n ) {
    // do something
    // increment counter
    i = i + 1;
}
```

### Pattern for doing something an indefinite number of times

```
% initialization

while ( not stopping signal ) {
    // do something
    // update status (variables)
}
```

### Shortcut expressions

- Increment: $i++$
- Decrement: $i--$
- Assignment operators:
  - $s += val$
  - $s -= val$
  - $s *= val$
  - $s /= val$
**Example 1: Eeeeeeeeeee!**

The exponential function can be approximated by the series \( e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots + \frac{x^n}{n!} \). One expects that the approximation is “better” when more terms in the series are used.

We will use method `Math.exp()` to calculate the “true” value of \( e^x \) and attempt to determine “how good” the above series approximation is. The difference between the true value and the approximation is the **error**. When we approximate, the amount of error that we are willing to tolerate is called the **tolerance**.

Write a program segment that starts by approximating \( e^x \) by just the first term of the series and then add one term at a time until a tolerance of 0.0001 is satisfied. \( x \) is to be input by a user.

```java
System.out.print("Enter power of e: ");
double x = Keyboard.readDouble();
double ans = Math.exp(x);  // true value of e^x
double ex = 1;             // approx value of e^x so far
double tol = 0.0001;       // error tolerance

for (int k = 1; Math.abs(ans-ex) > tol; k++)
    System.out.print("Error after " + k);
System.out.println(" terms:  "+ Math.abs(ans-ex));
```

**Example 2: Count down**

Write a program segment to print “count-down messages.” User enters the number of seconds to go (a positive integer). E.g., if user enters 3, display the messages

- T-3 seconds
- T-2 seconds
- T-1 second
- Take-off!

Use the `for` loop.

```java
int t = Keyboard.readInt();  // time left

System.out.println("T-1 second");
System.out.println("Take-off!");
```