Topics: iteration (while loop), CW demo

Reading (JV): Sec 3.6

The while loop

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

Pattern for doing something \( n \) times

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

Pattern for doing something an indefinite number of times

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

Example 1: \( n \) factorial

Write a program segment for calculating \( n! \). Assume \( n \) is given. Use a while loop.

Shortcut expressions

Increment: \( i++ \);
Decrement: \( i-- \);
Assignment operators: \( s += val; \)
\( s -= val; \)
\( s *= val; \)
\( s /= val; \)
Example 2: Eeeeeeexxeee!

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 0.001 is satisfied. \( x \) is to be input by a user.