Previous Lecture:
- Variables and their scope
- Method call and execution
- A simple loop

Today’s Lecture:
- for loop
- Defining a class—writing methods

Assigned reading:
- T Sec 3.1.1, 3.1.2

The for loop
Pattern for doing something \( n \) times:

```java
for (int i=0; i<n; i++) {
    // do something
}
```

Shortcut expressions
- Increment operator
  \( i = i + 1; \Rightarrow i++; \)
- Decrement operator
  \( i = i - 1; \Rightarrow i--; \)

Use them on their own—not as part of another statement!

More shortcut expressions
- Assignment operators
  \( s = s + val; \Rightarrow s += val; \)
  \( s = s - val; \Rightarrow s -= val; \)
  \( s = s * val; \Rightarrow s *= val; \)
  \( s = s / val; \Rightarrow s /= val; \)

Another example:
- Write a program fragment to approximate the area of a circle using \( n \)-gons. What should \( n \) be?

Area of \( n \)-gon

Inscribed hexagon: \( (n/2) \sin(2\pi/n) \)

Circumscribed hexagon: \( n \tan(\pi/n) \)
The for loop

Pattern for doing something *an indefinite number of* times:

```java
for ( initialization; not stopping signal; update ) {
    // do something
}
```

Method with input parameter

- Write an instance method `expand(double f)` that expands the `Interval` by a factor of `f`.
- Where do you put this method?
- What should be the method header?

- Parameter of *primitive type*: pass by value
  I.e., value is copied

```java
/** Numeric interval */
class Interval {
    private double base; // low end
    private double width; // interval width

    /** Getter method */
    public double getEnd() {
        return base + width;
    }

    /** Setter method */
    public void setWidth(double w) {
        width = w;
    }
}
```

```java
public class Client {
    public static void main(String[] args){
        Interval i1 = new Interval();
        i1.setWidth(1.5);
        double x = 2;
        i1.expand(x);
        System.out.println(i1.getEnd());
    }
}
```