Method with input parameter

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

- Parameter of primitive type: \text{pass by value}
  I.e., value is copied

```java
public class Client {
    public static void main(String[] args) {
        Interval i1 = new Interval(0.2, 0.7);
        double x = 2;
        i1.expand(x);
        System.out.println(i1.getEnd());
    }
}
```

Method with input parameter

- Write an instance method
  \[ \text{isIn} (\text{Interval } i) \]
- that returns the \text{boolean} value \text{true} if the instance is in \text{Interval } i. Return \text{false} otherwise.

- Parameter of non-primitive type: \text{pass by reference}
  I.e., Reference is copied; object itself is not copied

- Be sure to read Sec 5.0
public class Client {
    public static void main(String[] args) {
        Interval i1 = new Interval(0.2, 0.7);
        Interval i2 = new Interval(
            Math.random(), 0.2);
        if (i2.isIn(i1))
            System.out.println("Interval i2 
                + "is in Interval i1.");
        else
            System.out.println("Interval i2 
                + "is not in Interval i1.");
    }
}

// Check if self is subset of interval i
public boolean isIn(Interval i) {
    return
        base>=i.base && getEnd()<=i.getEnd();
}

// Check if self is subset of interval i
public boolean isIn(Interval i) {
    boolean in = (base>=i.base &&
        getEnd()<=i.getEnd());
    if ( in==true )
        return true;
    else
        return false;
}

Class Variables & Methods

- Shared by all instances of a class
- Only one copy no matter how many objects have been instantiated
- Keyword: static
- Examples:
  - A variable to keep track of how many Intervals have been created
  - A constant used by the whole class
class Interval {
    private double base;  // low end
    private double range; // interval width
    public final static double maxWidth=5;

    // Constructor, range always <= 5
    public Interval(double b, double r) {
        base = b;
        range = Math.min(r, maxWidth);
    }

    // Other methods below ...
}

// Return overlapped Interval if
// Intervals a and b overlap

public class Client {
    public static void main(String[] args){
        Interval i1 = new Interval(0.2,0.7);
        Interval i2 = new Interval(Math.random(),0.2);

        Interval o = Interval.overlap(i1,i2);
    }
}
Method toString()

- Every object has default method toString
- Automatically invoked by print, println
  
  Interval a = new Interval(1,2);
  System.out.println(a);

- Some default text will be printed unless you define a toString method

Method overloading

- Different methods can have the same name
- A method has a signature: method name and the parameter types (including the order)
- In a class, all methods must have different signatures
- E.g., the abs method in the Math class

```java
class Interval {
    private double base; // low end
    private double range; // interval width
    public final static double maxWidth=5;

    public Interval(double base, double r){
        this.base = base;
        range = r;
    }

    public String toString() {
        return "[" + base + ",", + getEnd() + "]";
    }
}
```