- Previous Lecture:
  - Defining a class:
    - Constructors
    - Keyword this
    - Executing an instance method
- Today's Lecture:
  - Defining a class:
    - Non-primitive type parameters and return
    - Static variables and methods
    - Method overloading
- Reading: Sec 9.10, 9.11, Sec 8.1

Static Variables & Methods

- *Shared* by all instances of a class
- Only one copy no matter how many objects have been instantiated
- Keyword: **`static`**
- Examples:
  - A constant used by the whole class
  - A variable to keep track of how many Intervals have been created
  - A method that doesn't need to reference the fields in objects

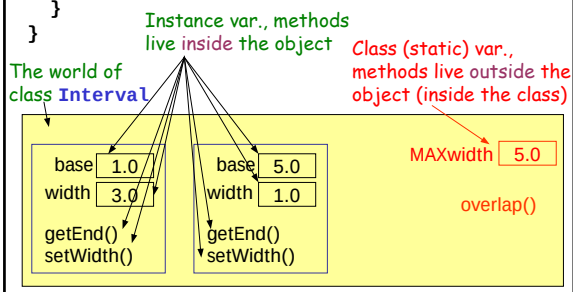
Class (`static`) method

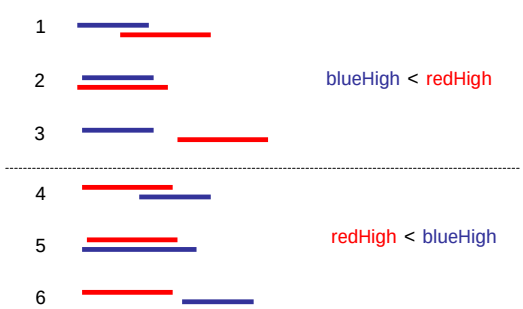
Write a class method
`overlap(Interval a, Interval b)`
that returns a new **`Interval`** representing the overlap between **`Interval`**s **`a`** and **`b`**. (Return **`null`** if there's no overlap)

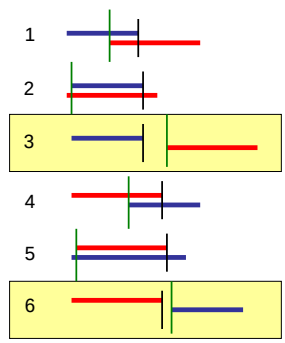
Where will the method live?

What should be the method header?


```
public class Client {
  public static void main(String[] args) {
    Interval i1= new Interval(1,3);
    Interval i2= new Interval(5,1);
  }
}
```







The overlap's left is the rightmost of the two original lefts

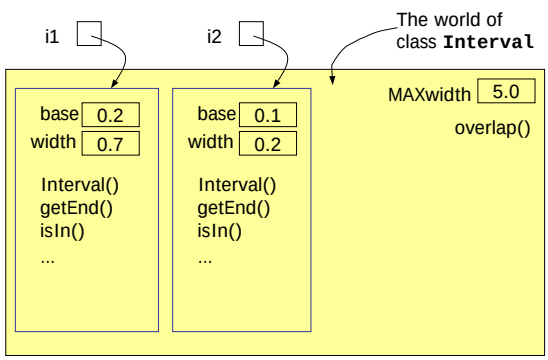
The overlap's right is the leftmost of the two original rights

No overlap if OLeft>ORight

```
/* =the overlapped Interval between
   Intervals a and b */
public static Interval overlap(Interval a,
                               Interval b) {
  Interval olap;      // overlapped interval
  double left, right; // olap's left & right

  left = Math.max(a.getBase(),b.getBase());
  right = Math.min(a.getEnd(),b.getEnd());
  if ( (right-left) <= 0 )
      olap= null;
  else
      olap= new Interval(left, right-left);
  return olap;
}
```



```
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);

  }
}
```

An instance `overlap` method

- Write an instance method
  `overlap(...)`
  that returns a new **`Interval`** if two **`Interval`**s overlap. Return **`null`** otherwise.
- What is the method header? What should be the parameters, if any?
- Are the static and instance versions very different?

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
- The return type is not part of the signature
- E.g., the **`abs`** method in the **`Math`** class


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

  public Interval(double b, double w) {
    setBase(b);
    setWidth(w);
  }

  public Interval() {}

  /* An Interval with base b and maxWdith */
  public Interval(double b) {
    setBase(b);
    setWidth(maxWidth);
  }
  // other methods below
```

`this(b,maxWidth)`