

- Previous Lecture:
 - Defining a class:
 - Constructors
 - Keyword `this`
 - Method `toString`
- Today's Lecture:
 - Defining a class:
 - Static variables and methods
 - Method overloading
- Reading: Sec 4.3, Sec 5.1
- Optional reading: Sec 5.2

November 3, 2005 Lecture 20 2

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

Instance var., methods live inside the object

The world of class `Interval`

Class (static) var., methods live outside the object (inside the class)

base 1.0 / width 3.0 base 5.0 / width 1.0 MAXwidth 1.0

getEnd() setWidth() getEnd() setWidth()

November 3, 2005 Lecture 20 11

```
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 double getEnd() {
        return base+width;
    }
    /* Set width to w, w<=MAXwidth */
    public void setWidth(double w) {
        width= Math.min(w,MAXwidth);
    }
}
```

November 3, 2005 Lecture 20 12

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 fields

November 3, 2005 Lecture 20 13

Class (static) method

Write a class method

```
overlap(Interval a, Interval b)
```

that returns a new `Interval` representing the overlap between `Intervals a` and `b`. (Return `null` if there's no overlap)

Where will the method live?
What is the method header?

November 3, 2005 Lecture 20 14

1 blueHigh < redHigh

2 blueHigh < redHigh

3 blueHigh < redHigh

4 redHigh < blueHigh

5 redHigh < blueHigh

6 redHigh < blueHigh

November 3, 2005 Lecture 20 17

1
2
3
4
5
6

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 $O_{Left} > O_{Right}$

November 3, 2005 Lecture 20 20

```

/* =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;
}

```

November 3, 2005 Lecture 20 23

```

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);
    }
}

```

November 3, 2005 Lecture 20 24

i1 i2 The world of class Interval

base 0.2 width 0.7
Interval()
getEnd()
isIn()
...

base 0.1 width 0.2
Interval()
getEnd()
isIn()
...

MAXwidth 5.0
overlap()

November 3, 2005 Lecture 20 25

An instance overlap method

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

November 3, 2005 Lecture 20 27

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

November 3, 2005 Lecture 20 29

```

class Interval {
    private double base; // low end
    private double width; // interval width
    public static final double maxWdith=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(maxWdith);
    }
    // other methods below
}

```

November 3, 2005

Lecture 20

30

Chain invocation of methods

- Suppose there are 3 intervals: `i1`, `i2`, `i3`
 - You know that `i1` and `i2` overlap
 - Write code to find if the overlapped interval of `i1` and `i2` is in interval `i3`
- ```

Interval i1 = new Interval(...);
Interval i2 = new Interval(...);
Interval i3 = new Interval(...);
// Assume i1 and i2 overlap
if (
 System.out.println("in i3");
else
 System.out.println("not in i3");

```

November 3, 2005

Lecture 20

32

```

Interval i1 = new Interval(...);
Interval i2 = new Interval(...);
Interval i3 = new Interval(...);
/* Without assuming that i1 and i2
 overlap */

```

November 3, 2005

Lecture 20

34

## A different example

- Create a `Person` class to organize data about a `Person`:
  - Name
  - Age
  - ...

November 3, 2005

Lecture 20

36

```

public class Person {
 private String name;
 private int age;

 public static final int LEGALage=18;
 /** Constructor */
 public Person(String name, int age)
 { this.name= name; this.age= age; }

 /** =This Person is an adult */
 public boolean isAdult()
 { return age >= LEGALage; }

 /** =String description of this Person */
 public String toString()
 { return name + " is " + age; }
}

```

November 3, 2005

Lecture 20

37

## Modify `Person` class

- Modify `Person` class to store data about a `Person`'s best friend: add another instance variable `friend`
- What should be the type of the field `friend`?
- Add two more methods to the class definition: `makeFrend`, `beFriendOf`

November 3, 2005

Lecture 20

38