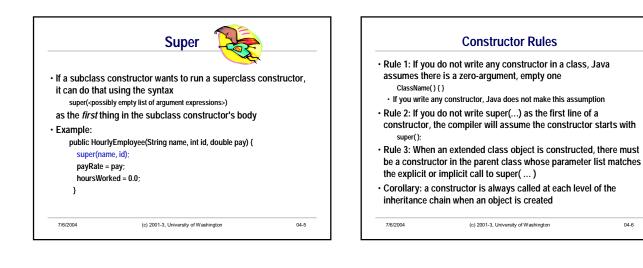


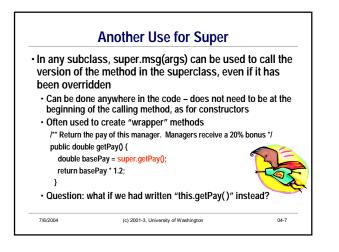
(c) 2001-3, University of Washington

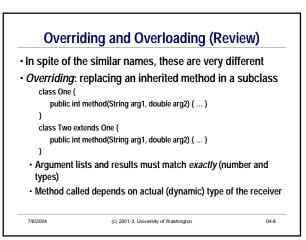
04-3

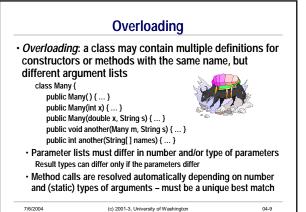
<text><text><text><code-block><text></code>

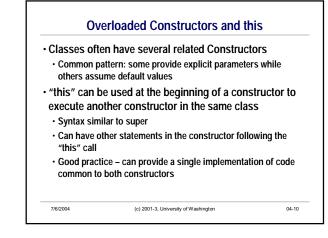
7/6/2004



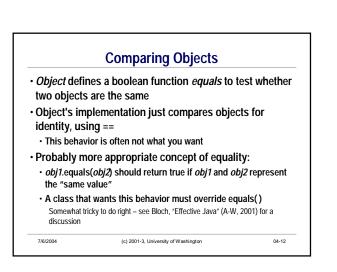


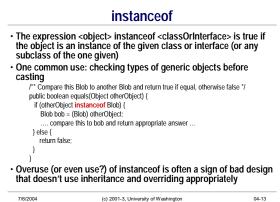


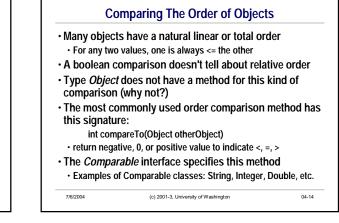




/** Construct an hourly employee with name, id, and pay rate */			
public HourlyEmployee(String name, int id, double pay) {			
super(name, id); payRate = pay;			
			hoursWor
}			
// default pa	y for new hires		
private stat	ic double defaultPay = 17.42;		
/** Construe	ct an hourly employee with name, id, and defa	ult pay rate */	
public Hour	lyEmployee(String name, int id) {	.,	
this(name	, id, defaultPay);		
}	3		







Sorting Lists of Objects

- Collections.sort(aList)
- Arrays.sort(anArray)

7/6/2004

- These two static methods can sort any objects, as long as...
- $\boldsymbol{\cdot}$ all list elements are Comparable (in the Java sense) and
- all list elements really are comparable (in the usual sense)
- This is one motivation for giving classes a compareTo method

(c) 2001-3, University of Washington

and using "implements Comparable"

04-15

Beyond compareTo What if you want to sort object is some way other than their defined compareTo? • Example: sort strings by length rather than alphabetical order

- What if you want to sort objects which are not Comparable?
 Example: sort Color objects
- Possible solutions: all either inconvenient, undesirable, or impossible

(c) 2001-3, University of Washington

- · change existing compareTo if it exists
- add compareTo if it doesn't exist
- make a subclass and override compareTo
- write your own code for sorting

7/6/2004

04-16

Comparator Objects

• A Comparator is an interface defining one method:

• int compare (Object obj1, Object obj2);

7/6/2004

returns value similar to compareTo, depending on whether obj1< obj2, ob1 equals obj2, or obj1 > obj2

• There are versions of Collections.sort and Arrays.sort which let you pass in a Comparator as a parameter.

- The compare method of the Comparator is written to perform exactly the kind of comparison needed
- Completely independent of the objects' compareTo methods.

(c) 2001-3, University of Washington

04-17

Copying Object and clone() [skipped Summer 2004]

- Review: what does A = B mean? (Hint: draw the picture)
- · This behavior is not always desirable
- In Java, the = operator cannot be overridden
- Instead, a method to copy can be written
- obj.clone() should return a copy of obj with the "same" value
 Object's implementation returns a new instance of the same class whose
 - instance variables have the same values as obj
- $\boldsymbol{\cdot}$ Object's implementation is protected
- If a subclass needs to do something different, e.g. clone some of the instance variables too, then it should override clone()

(c) 2001-3, University of Washington

clone cannot be used at will...

7/6/2004

· Class must be marked as "Clonable"

04-18

