

CSSE 220 Day 20

Inheritance recap
Object: the superest class of all
Inheritance and text in GUIs

Check out *Inheritance2* from SVN

Questions?

Exam 2 is on Monday, Oct 22, 2012 (7 – 9 PM)
Section 1: Olin O259

Project Team Preference Survey

- ▶ On ANGEL, under Lessons → Assignments
- ▶ Preferences help me to choose teams; I also consider your performance so far in the course
- ▶ Complete the survey by Monday, Oct 22, 2012, noon
- ▶ Most teams will have 3 students
- ▶ Are you willing to be on a team of 2
- ▶ List up to 5 students you'd like to work with, highest preference first.
 - You may not get your first choices, so it's a good idea to list more than two
 - Best to choose partners whose commitment level and current Java coding/debugging ability is similar to yours
- ▶ List up to 2 students you'd prefer NOT to work with
 - I'll do my best to honor this, but I must find a team for everyone.

1, Object

»» The superest class in Java

Object

- ▶ Every class in Java inherits from **Object**
 - Directly and explicitly:
 - **public class String extends Object {...}**
 - Directly and implicitly:
 - **class BankAccount {...}**
 - Indirectly:
 - **class SavingsAccount extends BankAccount {...}**

Object Provides Several Methods

- ▶ **String toString()**  Often overridden
- ▶ **boolean equals(Object otherObject)**
- ▶ **Class getClass()**  Sometimes useful
- ▶ **Object clone()**  Often dangerous!
- ▶ ...

Overriding toString()

- ▶ Return a concise, human-readable summary of the object state
- ▶ Very useful because it's called automatically:
 - During string concatenation
 - For printing
 - In the debugger
- ▶ `getClass().getName()` comes in handy here...

Overriding equals (Object o)

- ▶ Should return true when comparing two objects of same type with same “meaning”
- ▶ How?
 - Must check types—use **instanceof**
 - Must compare state—use **cast**
- ▶ Example...

Polymorphism

»» Review and Practice

Polymorphism and Subclasses

- ▶ A subclass instance is a superclass instance
 - Polymorphism still works!
 - **BankAccount ba = new SavingsAccount();**
ba.deposit(100);
- ▶ But not the other way around!
 - **SavingsAccount sa = new BankAccount();**
sa.addInterest();
- ▶ Why not?



BOOM!

Another Example

- ▶ Can use:

- `public void transfer(double amt, BankAccount o){
 this.withdraw(amount);
 o.deposit(amount);
}`

in BankAccount

- ▶ To transfer between different accounts:

- `SavingsAccount sa = ...;`
- `CheckingAccount ca = ...;`
- `sa.transfer(100, ca);`

Summary

- ▶ If B extends or implements A, we can write

`A x = new B();`

Declared type tells which methods x can access.
Compile-time error if try to use method not in A.

The actual type tells which class' version of the method to use.

- ▶ Can cast to recover methods from B:

`((B)x).foo()`

Now we can access all of B's methods too.

If x isn't an instance of B, it gives a run-time error (class cast exception)

BallWorlds



- Meet your partner
- Carefully read the requirements and provided code
- Ask questions (instructor and TAs).

BallWorlds Teams – Section 1

csse220-201310-BW10, crumpaa, fullerga
csse220-201310-BW11, leversad, llewelsd
csse220-201310-BW12, sneedbj, zajacrc
csse220-201310-BW13, boucheka, earlda
csse220-201310-BW14, heibelcj, hortoncb
csse220-201310-BW15, evansda, hiancejk
csse220-201310-BW16, goldsbge, yinm
csse220-201310-BW17, huangf, puhrijj
csse220-201310-BW18, ametsid, quj
csse220-201310-BW19, wangl2, winterc1

Check out *BallWorlds* from SVN

BallWorlds Worktime

»» Pulsar, Mover, etc.

You can turn BallWorlds in on Monday before 11:59 PM for full credit. If you miss that deadline, you may turn it in by Tuesday at 11:59 p.m. for 90% credit.