COP 3330 2/27/26

1. Watch Wed Video (it's long)

2. End of Class - Hand Back Quizzes

3. Cur Grade/Perc $= \left((P1+P2+P3)/10 + (Q1+Q2)\overset{*}{\,}5 \right)/.31$

   ~ A = 80%, B = 64%, C = 45%

   Bimodal a little bit.

4. TODAY'S LEC - POLYMORPHISM

   LOOK AT MECHANICS

   POLY → many, MORPH → forms

   ~~Refern~~ Reference Type and Object Type need not be identical.

int y → B extends A ← int x

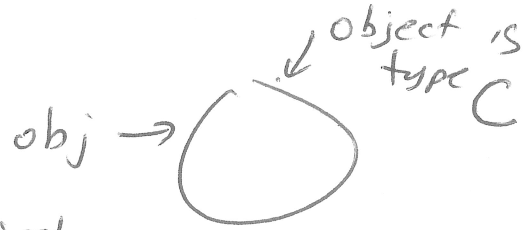
int z → C extends B

methods called f, g take in reference type A, B and C.

Difficulty is figuring out which method will get called when multiple methods "fit" the given method call.

Rule #1: **this** gets binded dynamically at run time

A obj = new C(--)



obj.f(...) → Java first figures out that obj is pointing to an object of ~~type~~ class C and looks for an appropriate method in class C.
(If C doesn't have it, then we look at B, then A, --)

Rule #2: for parameters we use reference type.

mine.f(obj)

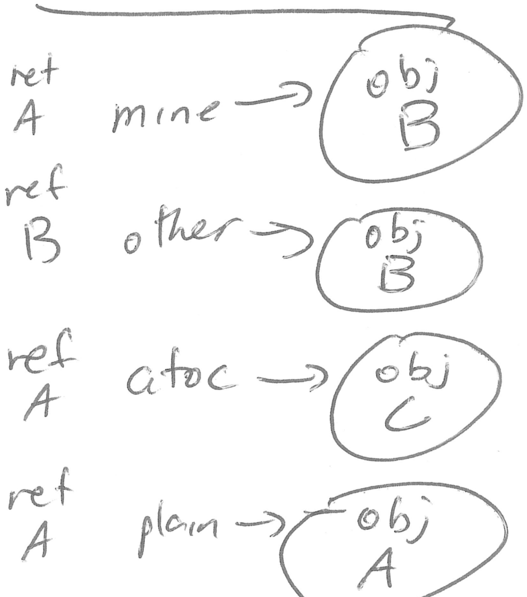
reference type is A, so this is what we'll use to find a matching method

Class of object mine is pointing to

addendum but I could do

if obj instanceof C:
((C)obj).methodInC(...)

Example 1



Ref		Obj
A	→	A, B, C
B	→	B, C
C	→	C

an A is NOT a C
but a C is-A A.

OBJ B REF A

* * why does this print out f(a)?

other.f(mine) → calls x.g()

OBJ B REF B

mine.f(other)

OBJ B

mine.f()

CAST OBJ B REF B

* *

((B)mine).f(other)

OBJ C REF A

atoc.f(plain) → x.g() is on OBJ A

OBJ C REF A

atoc.f(atoc) → no match C or B f(a)
→ x.g()

Output

f(a)
g(b) test 1

f(a)
g(b) test 2

~~f(b)~~

test#3 In default f(b)

f(b)
g(b) test 4

f(a)
g(a) test5

f(a)
g(c) test6