Outline

Polymorphism

- Terminology
- Pointers, references, "cutting"

Base and derived objects in memory

Vtables

Logistics

- Disc Tomorrow
- Project posted due Sunday 11:59 pm
- Friday : peer mentor presentation

  Exit evaluation (not graded)

  Feedback Survey

practice

- Saturday Final

Polymorphism : void f (Animal &a) {...}

f (Dog (~))   vector<Animals>

Inheritance : class Derived : public Base

- When you use a double for an int
  kind of Polymorphism

Inheritance is when you define a Derived class
that "is a" Base class

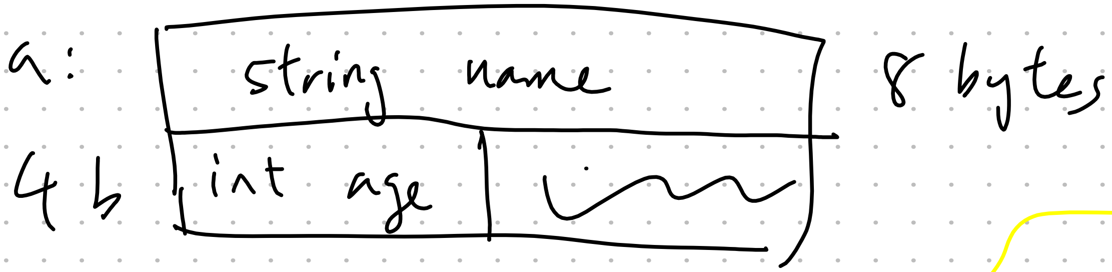
Another way of achieving Polymorphism w/o Inheritance
is Interfaces / Traits

Duck can Float   BeachBall can Float
→ not "isa Float"

f(Float f) { dunk f in water}

Objects in Memory

Animal a:

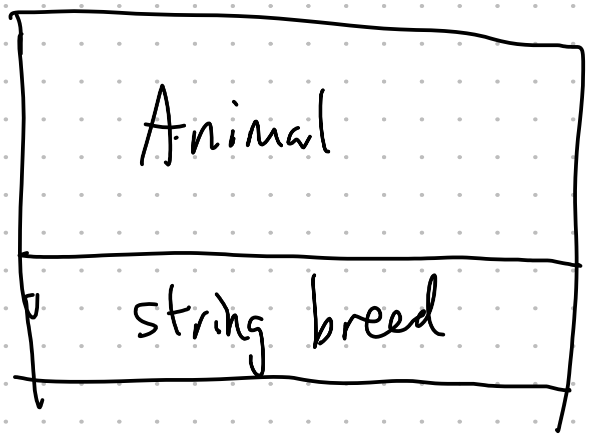


assuming no virtual fns

Dog d;

Dog *p = &d

Animal *q = p;



p is convertible into q

Husky h

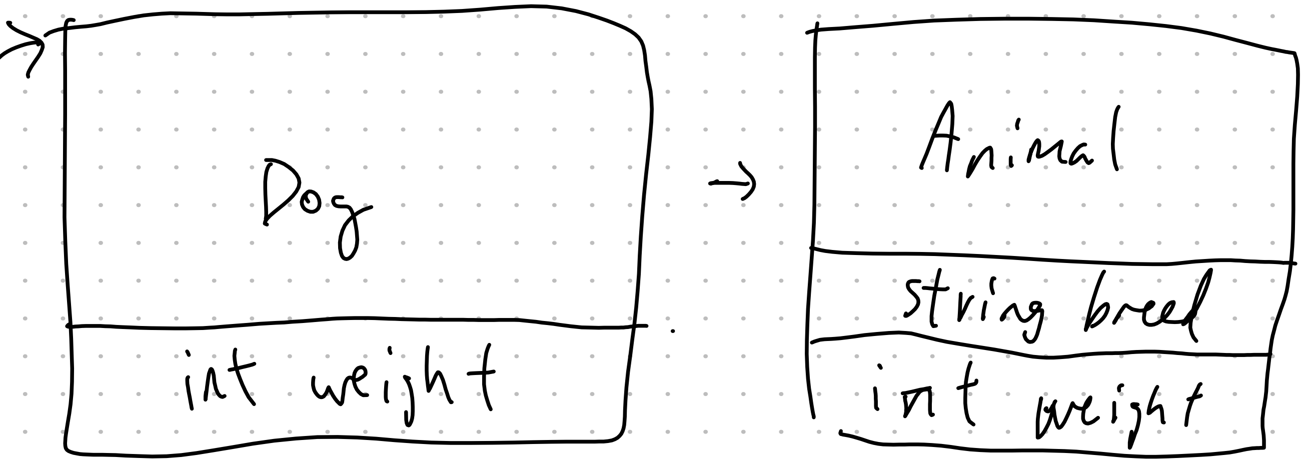
Husky *p = &h

↓

Dog * p = &h

↓

Animal *p = &h

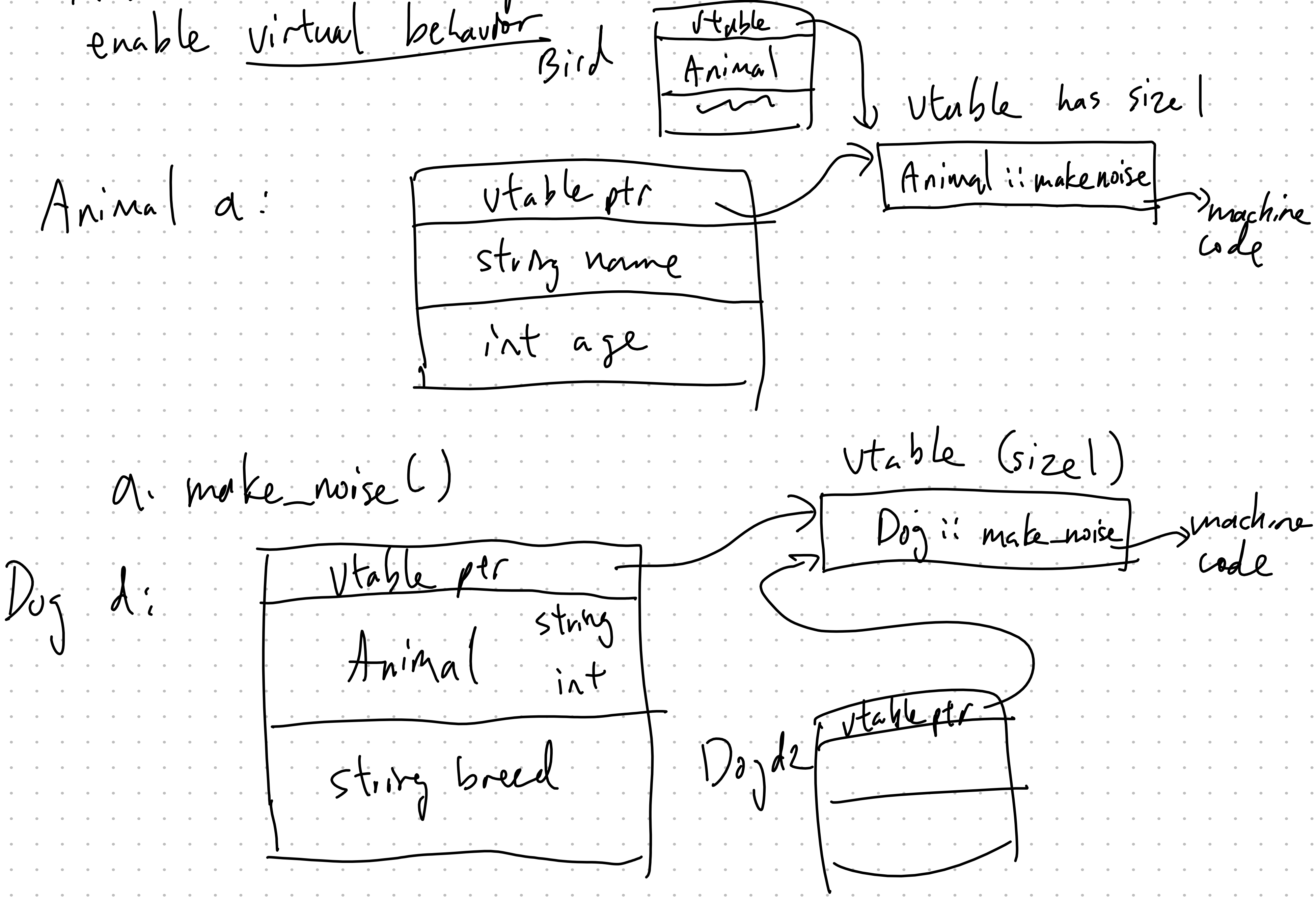


Husky &r = h;

Dog &r = h;

Virtual Fns are implemented via vtables

- Because make_noise is virtual, the compiler will create a vtable for all Animals & subclasses of Animal
- A vtable is an array of function pointers that enable virtual behavior



d.make_noise()

When we call a virtual method:

runtime steps

1. follow vtable ptr at the beginning of the obj to get to the correct vtable
2. Look up the virtual method in the vtable and call it

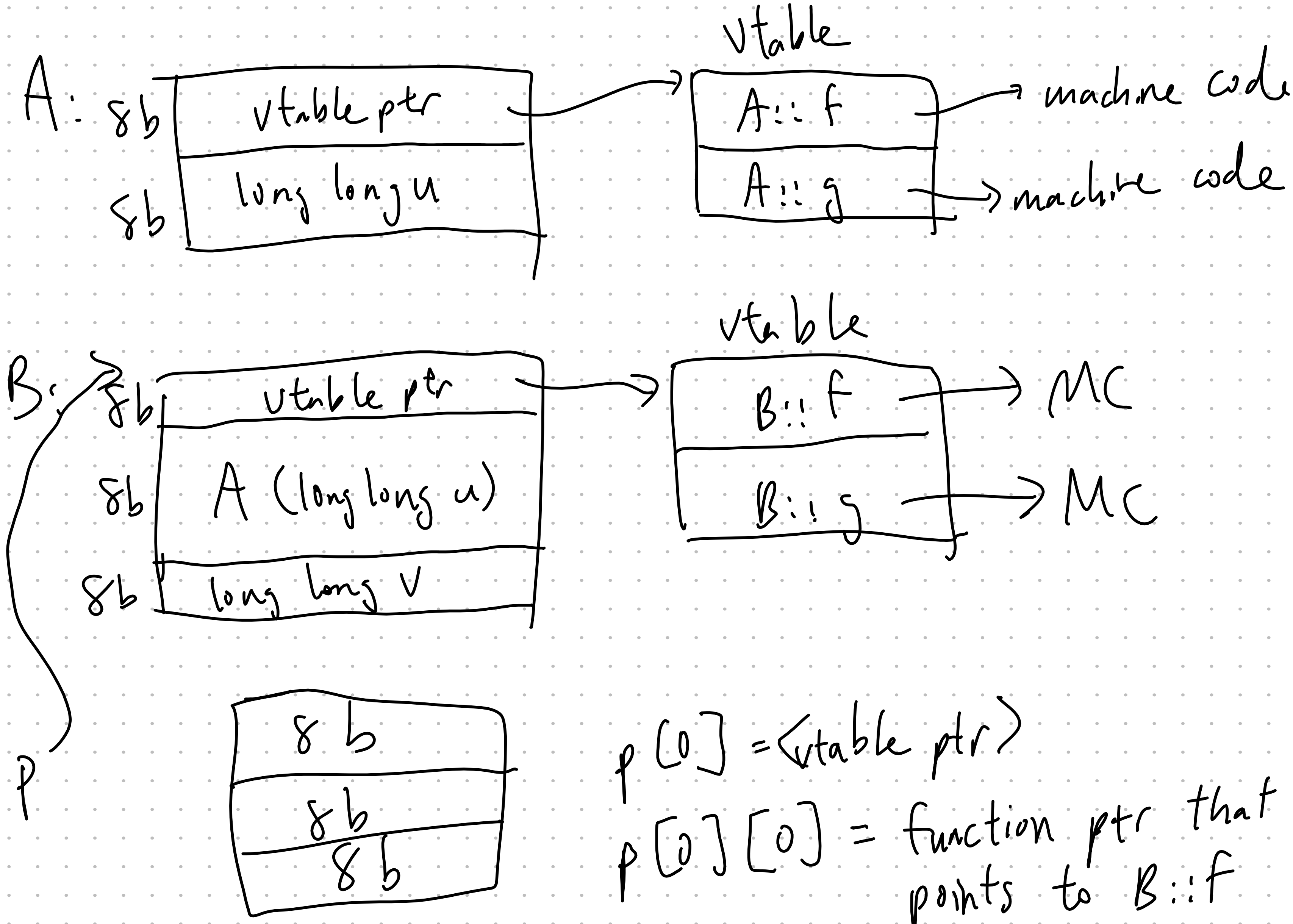
- 2 pointer dereferences could slow down program

- "Virtual behavior"

↳ Dynamic Dispatch

at runtime "deciding what method to call"

Static dispatch?



p[0] = <vtable ptr>

p[0][0] = function ptr that points to B::f

a method is just a function that takes in the this ptr as an implkit first argument.

UPE Tutoring

- UPE is the CS honor society
- Week 3 – week 9 M – F 9am – 5pm