Static Type and Dynamic Type. Dynamic Method Selection.

Static type a.k.a Compiled type

Dynamic type a.k.a Run-time type

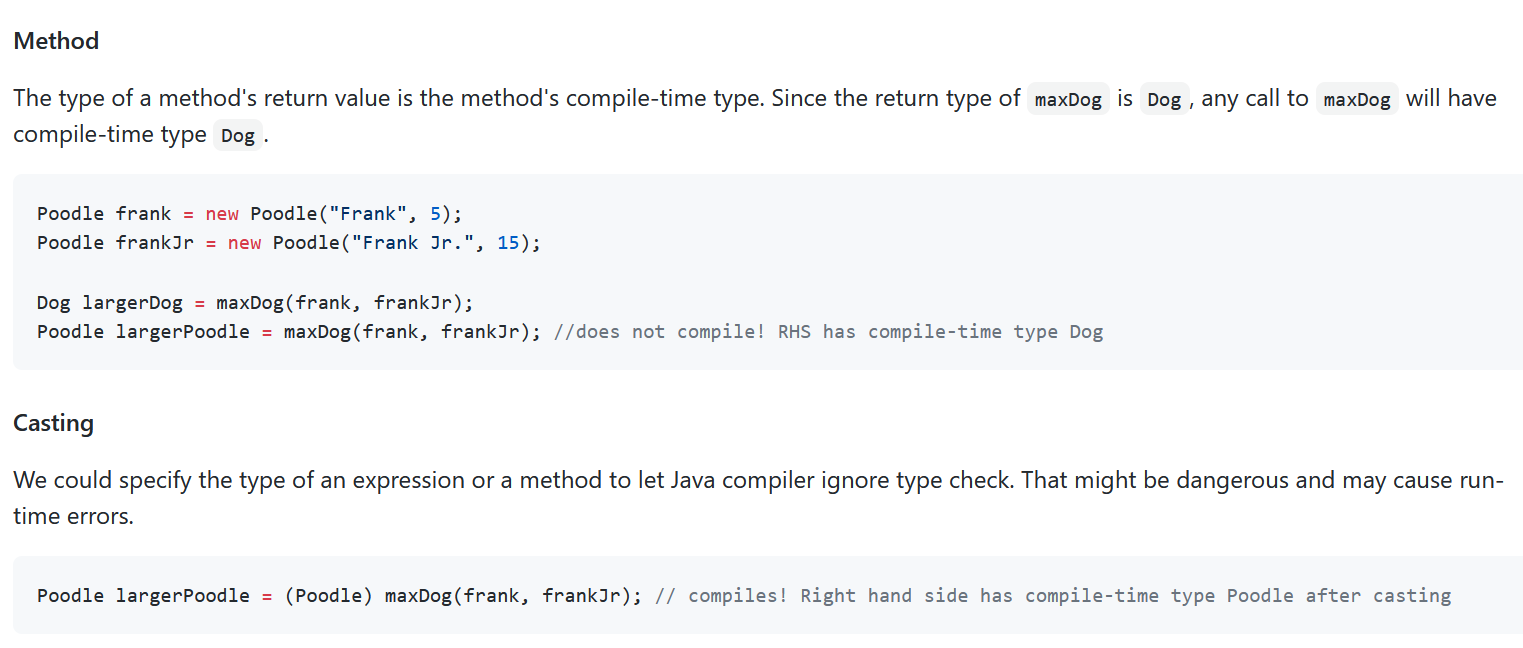
* Dynamic type needs to be under the category of the Static type

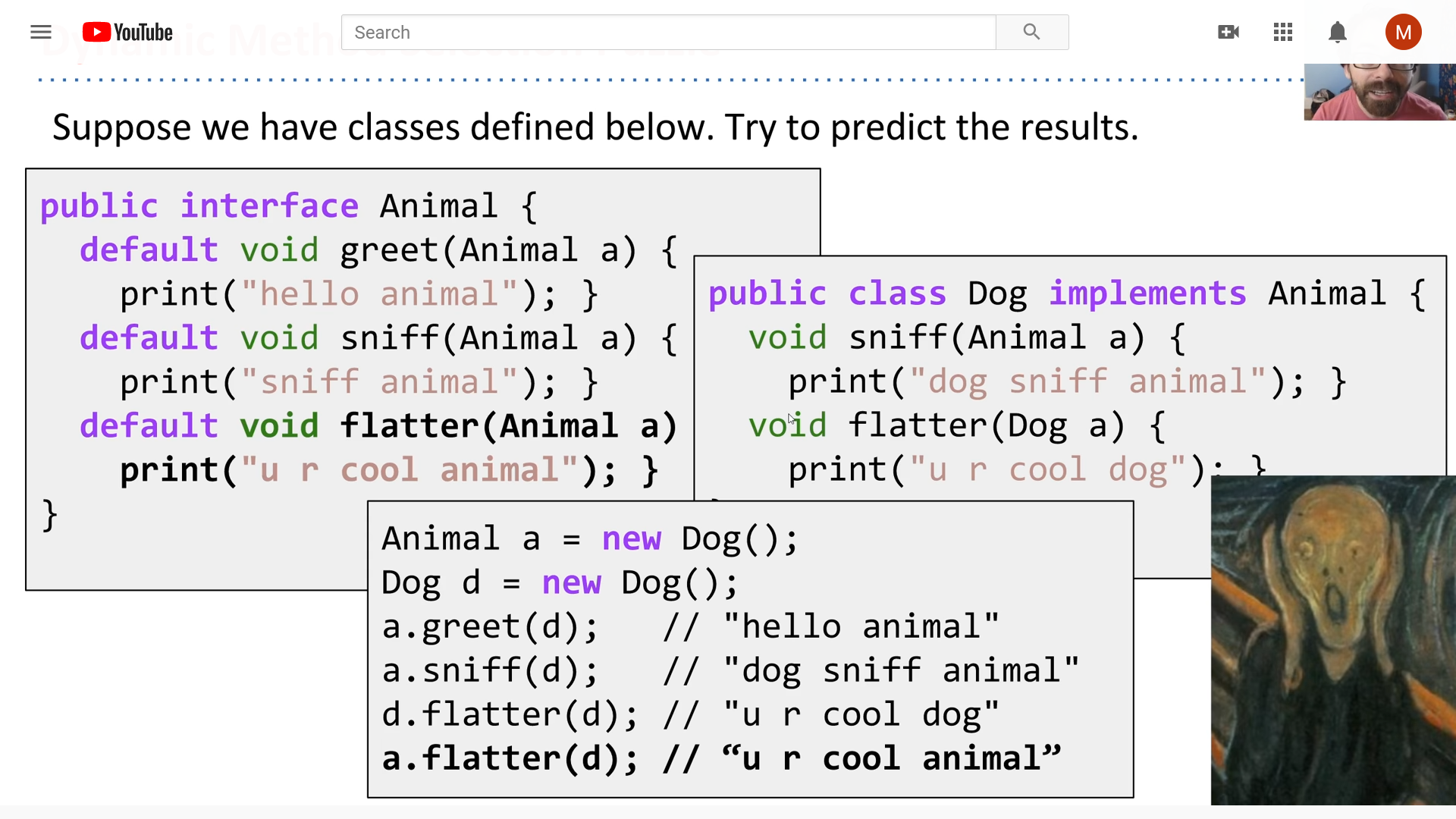
For example: Animial d = new Dog(); is okay

But Dog d = new Animal(); is not okay.

https://github.com/xiaoyang-liu-cs/berkeley-cs61b/blob/master/Lecture%20Notes/Week%2004.md







<https://www.youtube.com/watch?v=OHgKMipGdy8&list=PL8FaHk7qbOD6km6LlaHLWgRl9SbhlTHk2&index=9>

a’s static type Animal, Dynamic type Dog.

d’s static type and dynamic type are both Dog.

a.greet(d)

Okay, a, an Animal, do Animal has a greet method that can accommodate a type of Dog?

Yes. In Animal, we have greet(Animal a) and Dog is a type of Animal.

a.sniff(d)

Okay, a, an Animal, do Animal has a sniff method that can accommodate a type of Dog?

Yes. In Animal, we have sniff(Animal a) and Dog is a type of Animal.

However, Dog class also has a sniff method that can accommodate a type of Dog and this method overrides the sniff method in Animal. And a’s dynamic type is Dog, so Java executes the sniff method in Dog. This is called **Dynamic Method Selection.**

注意，如果in Animal there isn’t a method sniff that accommodates a Dog. It won’t turn to the sniff method in Dog and will end up with compilation error.

d.flatter(d)

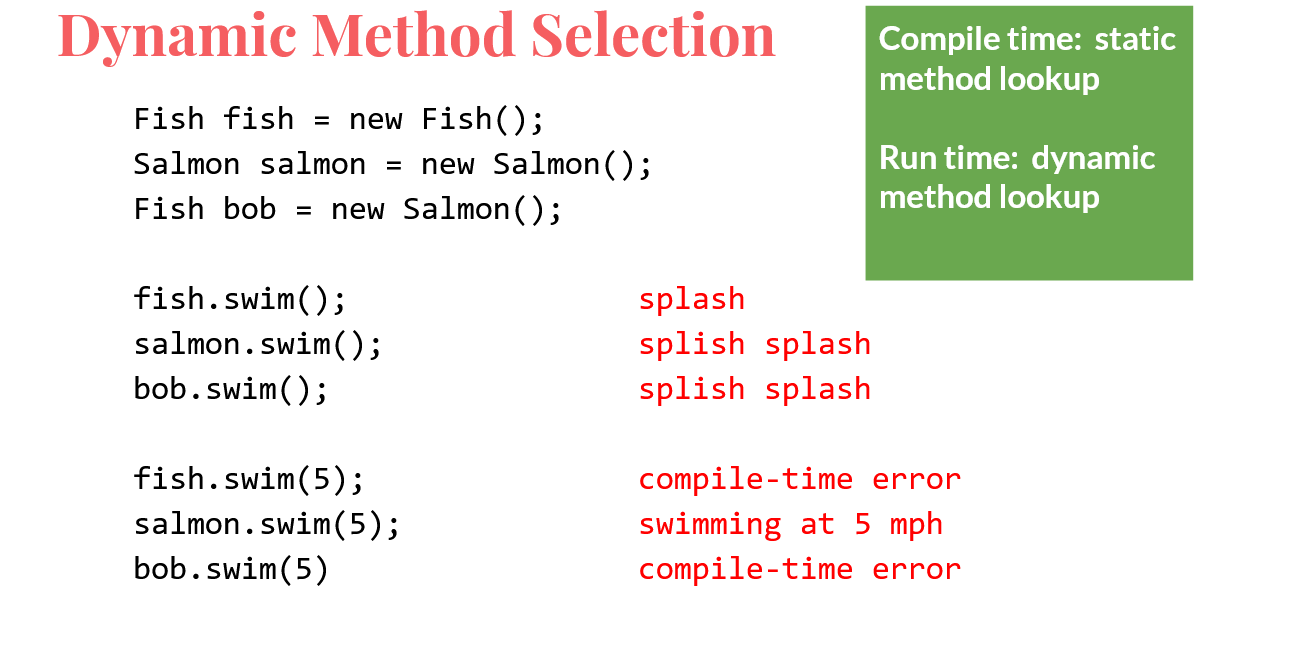
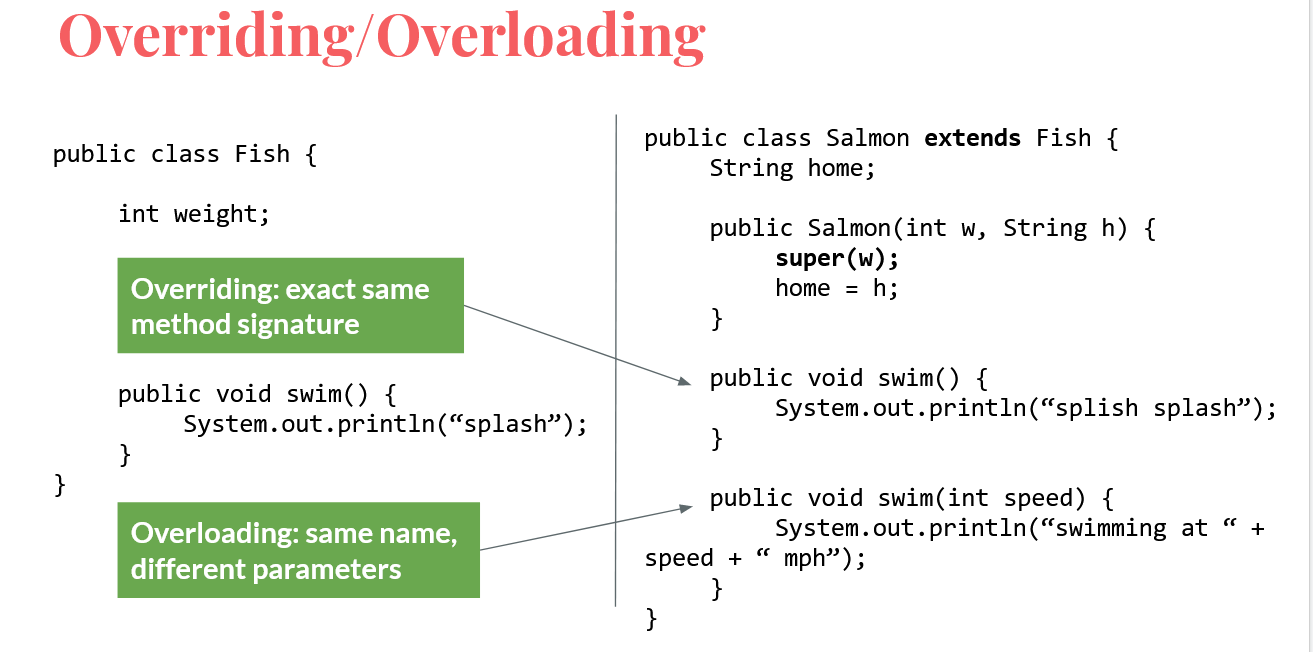
Okay, d, a Dog, does Dog has a flatter method that can accommodate Dog?

Yes. In Dog, we have flatter(Dog a) to handle this.

a.flatter(d)

Okay, a, an Animal, does Animal has a flatter method that can accommodate Dog and no overriding flatter method in Dog class? Yes, the flatter method in Dog class is not overriding because the signatures are different. The flatter method in Dog class is overloading. Therefore, a will go with the flatter method in Animal, not Dog.

Conclusion: an object will execute its static type’s class method, unless this method is overridden in dynamic class.



From this we know that: Static type of an object is important. Always look for the method in the static type first. If it does not exist, then compile- time error. If exists, run the method in the static type class. If it exists but it is overridden by a method of the object’s dynamic class, then execute the dynamic class’s method.