

LEC 11

CSE 123

Inheritance

Questions during Class?
Raise hand or send here

sli.do #cse123



BEFORE WE START

Talk to your neighbors:

What is your favorite summer activity?

[Respond on sli.do!](#)

Instructor: Trien Vuong

TAs:	Cora	Nhan
	Jonah	Nichole
	Maitreyi	Shiven

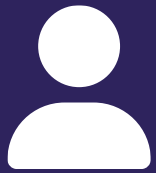
Music: [CSE 123 26su Lecture Tunes](#)

Announcements

- Quiz 2 is done!
 - Grades out before the final exam
- Programming Assignment 2 out tomorrow
- Creative Project 2 due tonight at 11:59 PM
- Resubmission Cycle 4 is out, due Friday (8/7) at 11:59 PM
- Make sure to check section participation and reach out to your Tas if you see anything off!

Lecture Outline: Inheritance

- Inheritance ◀



Practice : Think

[sli.do](#)[#cse123](#)

Which of the following are true? (Select all that apply.)

- A. If ClassA extends ClassB, then ClassA is the superclass
- B. Classes can extend multiple classes (e.g. "ClassA extends ClassB, ClassC")
- C. Java will prevent you from writing "ClassA extends ClassB" unless ClassA "is-a" ClassB
- D. You can call a method from the superclass on an instance of the subclass



Practice : Pair

[sli.do](#)

#cse123

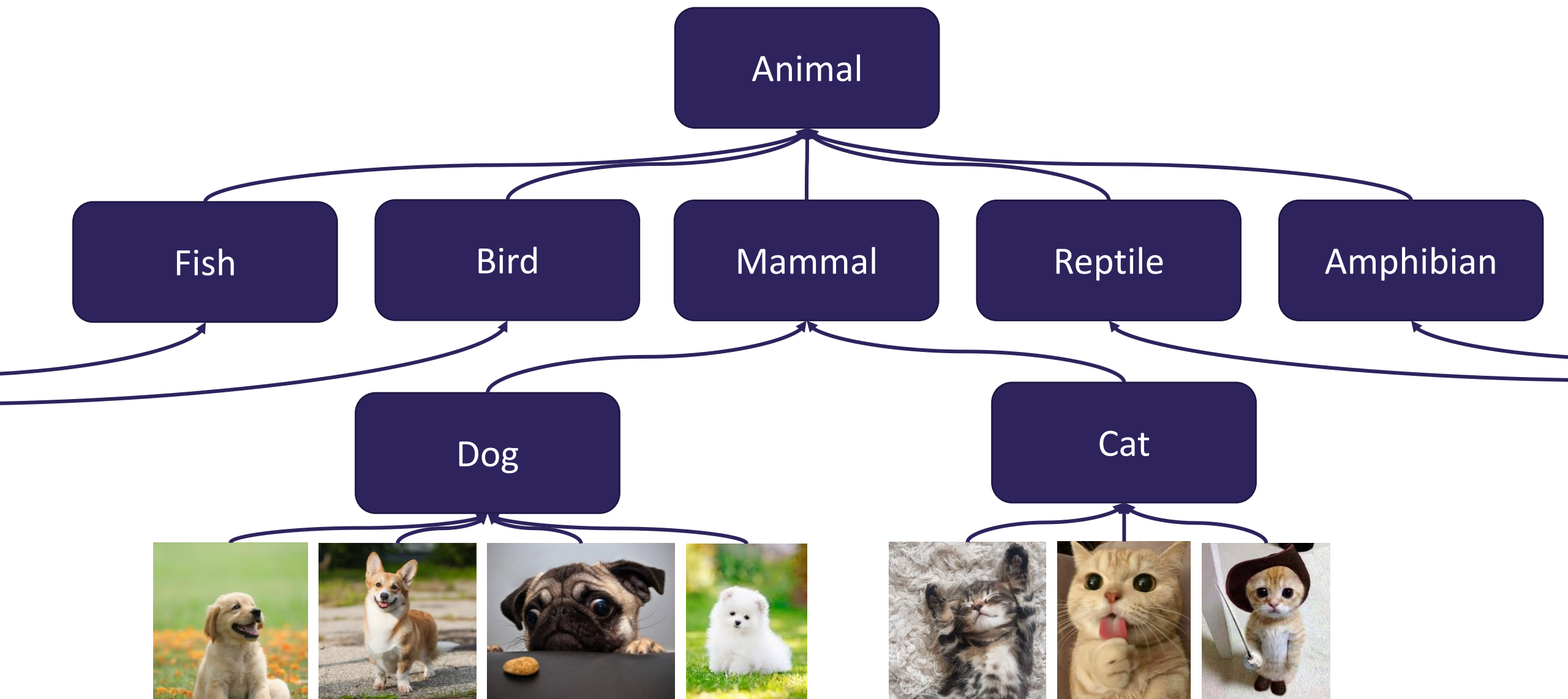
Which of the following are true? (Select all that apply.)

- A. If ClassA extends ClassB, then ClassA is the superclass
- B. Classes can extend multiple classes (e.g. "ClassA extends ClassB, ClassC")
- C. Java will prevent you from writing "ClassA extends ClassB" unless ClassA "is-a" ClassB
- D. You can call a method from the superclass on an instance of the subclass

Inheritance

- Connect together a “subclass” and “superclass”
 - Borrow / “inherit” code to reduce redundancy
 - `super()` keyword can be used just like `this()` (in a constructor)
- Syntax: `public class Subclass extends Superclass`
- *Should* Represent “is-a” relationships
 - `public class Chef extends Employee`
 - `public class Server extends Employee`
- In Java, all objects implicitly inherit from the `Object` class
 - `toString()`, `equals(Object)`, etc.

Is-a Relationships



The “is-a” Rule of Thumb

- class A should only extend class B if an A “is-a” B in the real world
- Java will *not* force you to follow this rule! That’s your job!

Inheritance Design: Typical Problem

- We give you:

Description of several classes, methods, and their behavior

- Your job:

Design an inheritance hierarchy that

- minimizes redundancy
- follows best practices (“is-a” test)

Inheritance Design: **Solution Technique**

- Use the “is-a” test to decide which classes should extend each other
- Keep an eye out for redundant code and consider whether you can reduce it even further using inheritance
 - Sometimes you can't, and that's ok!
 - The “is-a” test is more important than getting rid of all redundancy
- Often more than one reasonable answer!

What inheritance hierarchy did you come up with?