

# CS 113 – Computer Science I

## Lecture 17 – Class Design & Relationships

Thursday 03/28/2024

# Announcements

HW06 due next thursday

Lab06 **will not be autograded**. Also due next thursday. Get it checked off with the TAs

Exam Tuesday 4/23

Mid-semester feedback survey

<https://forms.gle/RytfNAYRUMQk24M86>

5 easy bonus points on your exam!! Complete by end of lab.

# Outline

- Review
- Access modifiers
- Inheritance

# Special methods

The **constructor method** is called when you do a `new`

**getters (aka accessors)**

return the values of instance variables

**setters (aka mutators)**

set the values of instance variables

**toString()**

returns a string representation of an object

# this

`this` is a special keyword that refers to the object inside an instance method

Allows us to access other instance variables within an instance method

# Access modifiers

Specify the access-level of instance variables/methods

- **public**
  - code outside of the class can access the variable/method
- **private**
  - code outside of the class cannot access the variable/method

Default in java is **public**

In this class, make instance data private

# Inheritance

# Bird Class

What properties does a bird have and what can it do?

- weight, color, fly

Let's write a bird class



# Lion Class

What properties does a lion have and what can it do?

- weight, color, run

Let's write a lion class

# Bird and Lion class

Duplicated code!

A Lion and a Bird are both *Animals*

Let's use **inheritance** to make this cleaner

# Inheritance

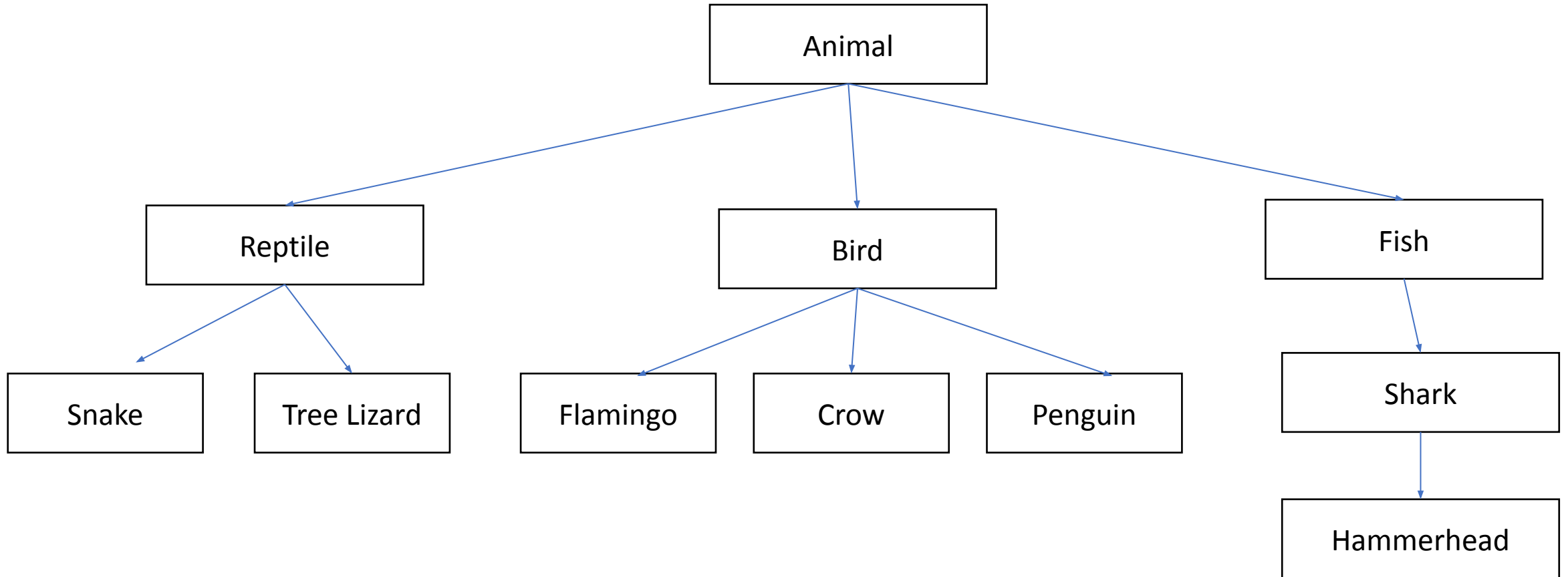
A class *inherits* variables and methods from an existing class.

The existing class is referred to as the **superclass** or **parent class**, and the new class is referred to as the **subclass** or **child class**.

Allows for code reuse

```
public class Bird extends Animal { ... }
```

# Inheritance: organizing classes into hierarchies



# Inheritance: subclasses refine behavior/state

Subclasses can override methods from parent class

# Exercise

Create a Zoo class with multiple animals

# Polymorphism

Program can treat all objects that extend a base class the same

Poly.java

Java decides at runtime which locomote to call

# Inheritance: constructors - `super()`;

We have some duplicated code between Animal and Bird constructors...

`super()`;

reference variable that is used to refer parent class constructor



# Inheritance: constructors - `super()`;

`super()`;

reference variable that is used to refer parent class constructors

Note:

`super`:

reference variable that is used to refer parent class object

# Example 2: Implement Minion

