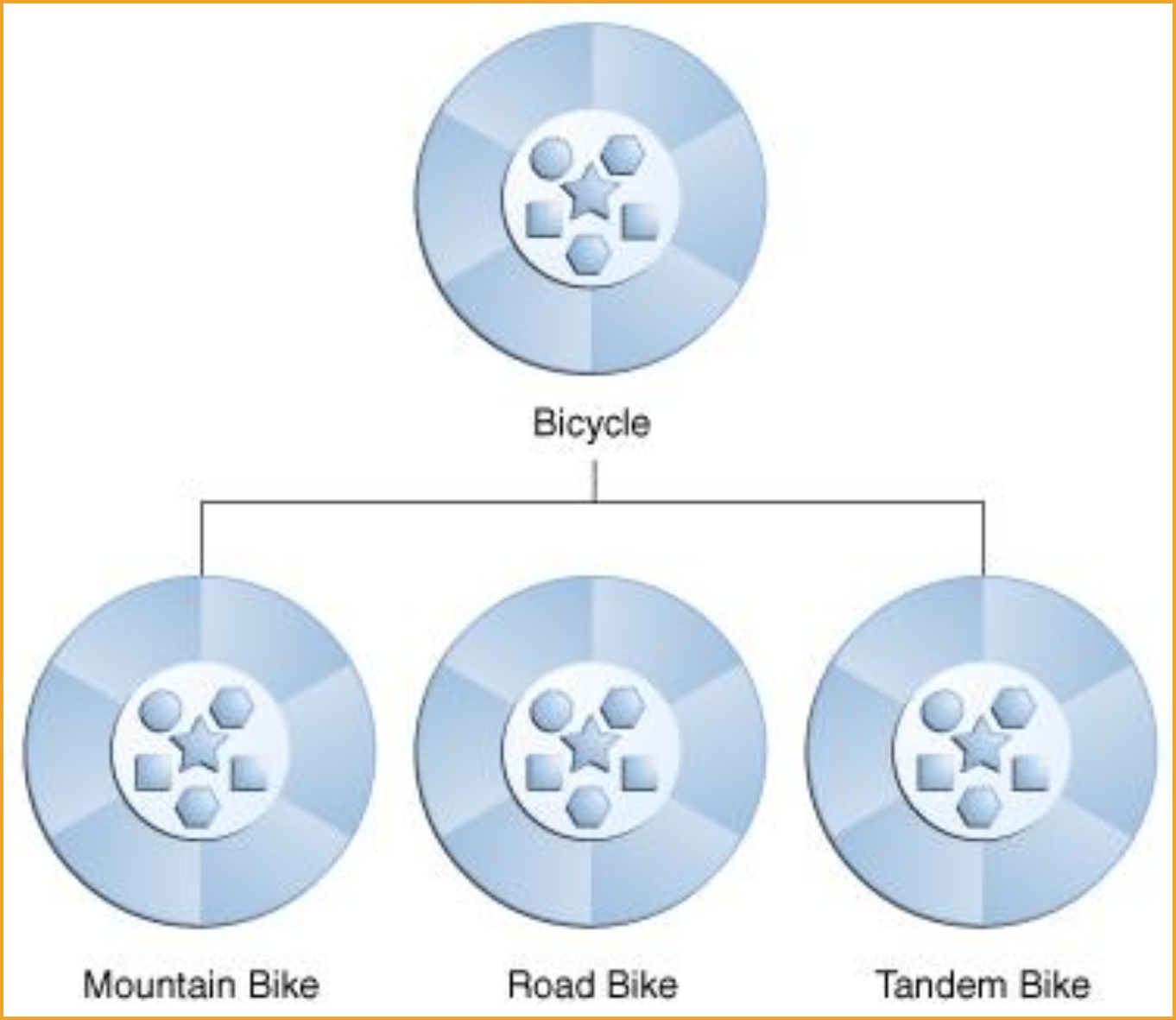
While you wait: look at this code and then answer the poll questions …

```
public class Animal {

    protected String name;

    public Animal(String name) {
        this.name = name;
    }

    public void move() {

        System.out.println(name +  " can move");

    }

    public String getName() {

        return this.name;

    }
}
```

```
public class Dog extends Animal {

    private String breed;

    public Dog(String name, String breed) {
        super(name);
        this.breed = breed;
    }

    public void move() {

        System.out.println(name + " can walk and run");

    }

    public void bark() {

        System.out.println("woof");

    }
}
```

```
Animal a1 = new Animal("fluffy");
Dog d = new Dog("rover", "poodle");
Animal a2 = d;
```



https://docs.oracle.com/javase/tutorial/java/concepts/inheritance.html

Inheritance

Inheritance and constructors (summary)

Java automatically generates a “no-args” constructor if you do not declare a constructor

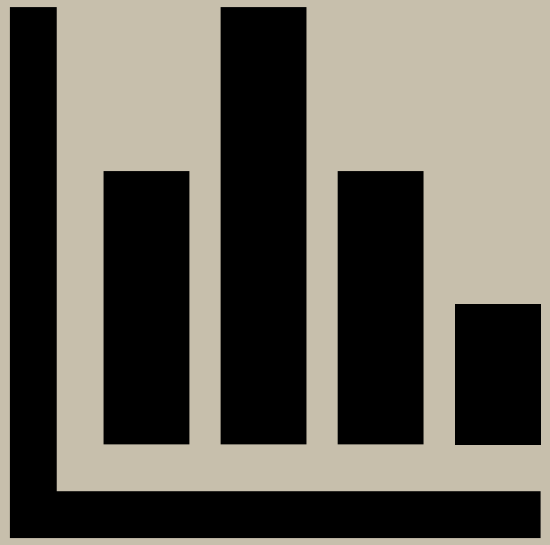
Any subclass constructor automatically tries to call the “no-args” constructor of the superclass, unless you explicitly call another constructor

THIS MEANS …

- If you are extending a class that only has constructors with parameters
- Then you HAVE TO write at least one constructor in your subclass
- And you HAVE TO call the appropriate superclass constructor in each subclass constructor

Inheritance and access modifiers (summary)

- Members declared **private** are not inherited by a subclass
- If you want something to be non-public but inherited, use **protected**
- A subclass can make a method **more** accessible, but not **less** accessible
- Static methods get **hidden**, not overridden, in a subclass



https://tinyurl.com/jp2-survey-week3

(Questions and answers will be posted on Moodle afterwards)