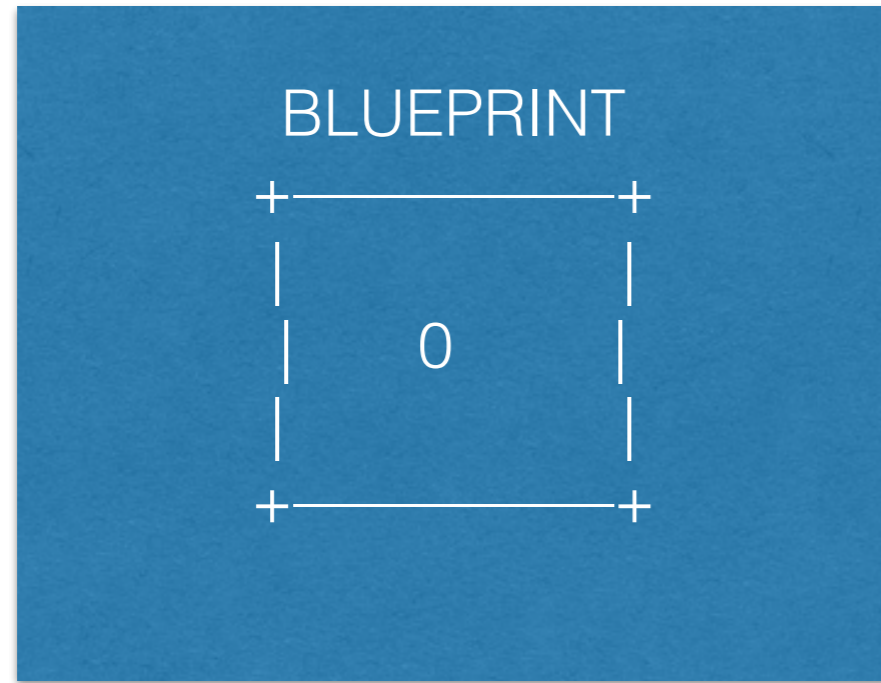


# Java, Classes, and Objects

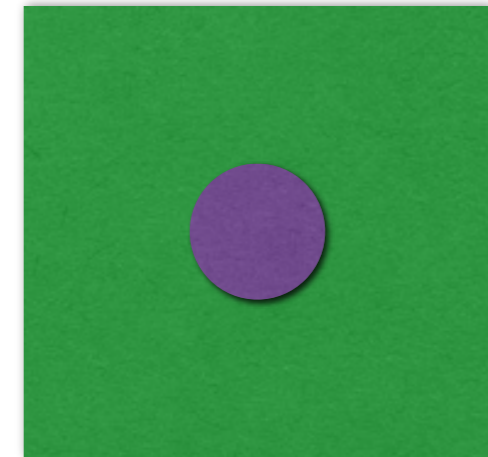
D. Thiebaut  
CSC212 — Fall 2014



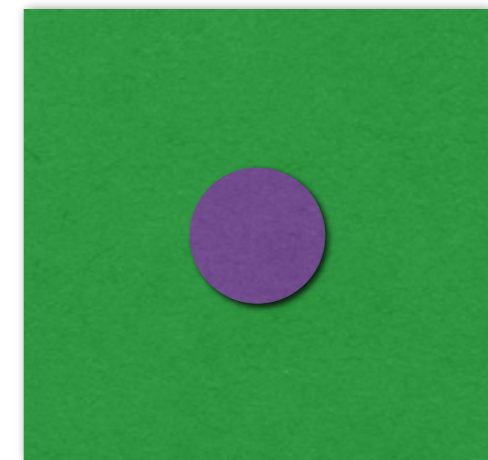
# Classes vs. Objects



Class



Object



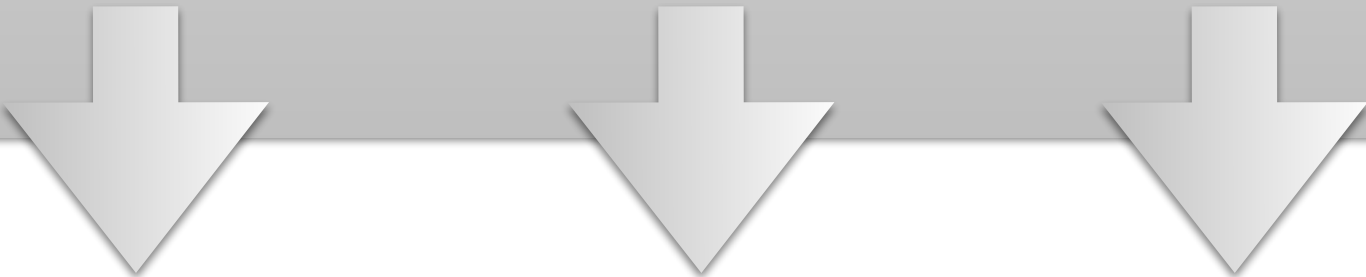
Object

# Review of Python Classes

(see Lab 1, Part 1)

# Example

```
class Person {  
    String name, Id;  
    int age;  
  
    Person( String n, int a, String i ) {  
        name = n; age = a; Id = i;  
    }  
  
    public void display( ) {  
        System.out.println( name + ": " + age + " years old" );  
    }  
  
    public static void main( String[] args ) {  
        Person p1 = new Person( "Sophia", 29, "990111222" );  
        p1.display();  
        Person p2 = new Person( "Max", 10, "990101010" );  
        p2.display();  
    }  
}
```



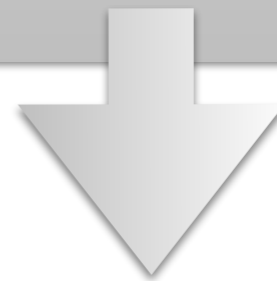
Sophia: 29 years old  
Max: 10 years old

# Important Concepts

- constructor
- method
- **new**
- overloading (see next slide!)

# Overloading

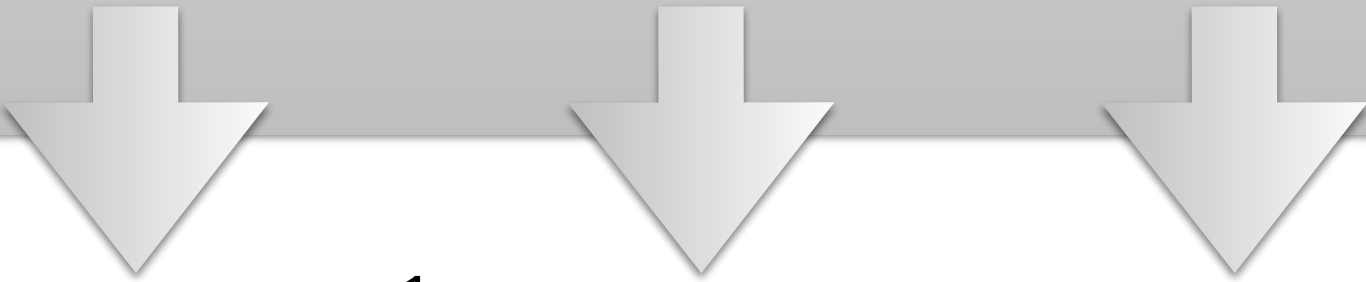
```
class Person {  
    String name, Id;  
    int age;  
  
    Person( String n, int a, String i ) {  
        name = n; age = a; Id = i;  
    }  
    Person( ) {  
        name = "Unknown"; age = 0; Id = "NA";  
    }  
    public void display( ) {  
        System.out.println( name + ": " + age + " years old" );  
    }  
    public static void main( String[] args ) {  
        Person p1 = new Person( "Sophia", 29, "990111222" );  
        p1.display();  
        Person p2 = new Person( );  
        p2.display();  
    }  
}
```



Sophia: 29 years old  
Unknown: 0 years old

# Overloading (2)

```
class Person {  
    // stuff removed  
  
    public void display( ) {  
        System.out.println( name + ": " + age + " years old" );  
    }  
    public void display( String caption ) {  
        System.out.print( caption + "\n==>" );  
        display();  
    }  
    public static void main( String[] args ) {  
        Person p1 = new Person( "Sophia", 29, "990111222" );  
        p1.display( "p1" );  
        Person p2 = new Person( );  
        p2.display( "p2" );  
    }  
}
```



p1  
==>Sophia: 29 years old  
p2  
==>Unknown: 0 years old

food



0	"Ice Cream"
1	"Sweet"
2	"Nutella"
3	"Chocolate"
4	"Mousse"
5	""

# Arrays

```
String[] food = new String[6];  
food[0] = "Ice Cream";  
for (int i=1; i<6; i++ )  
    food[i] = "";
```

or

```
String[] food = { "Ice Cream",  
                  "Sweet", "Nutella",  
                  "Chocolate", "Mousse",  
                  "" };
```



**The size of  
an array is  
FIXED!!!**



If your program accesses  
a cell outside the array...

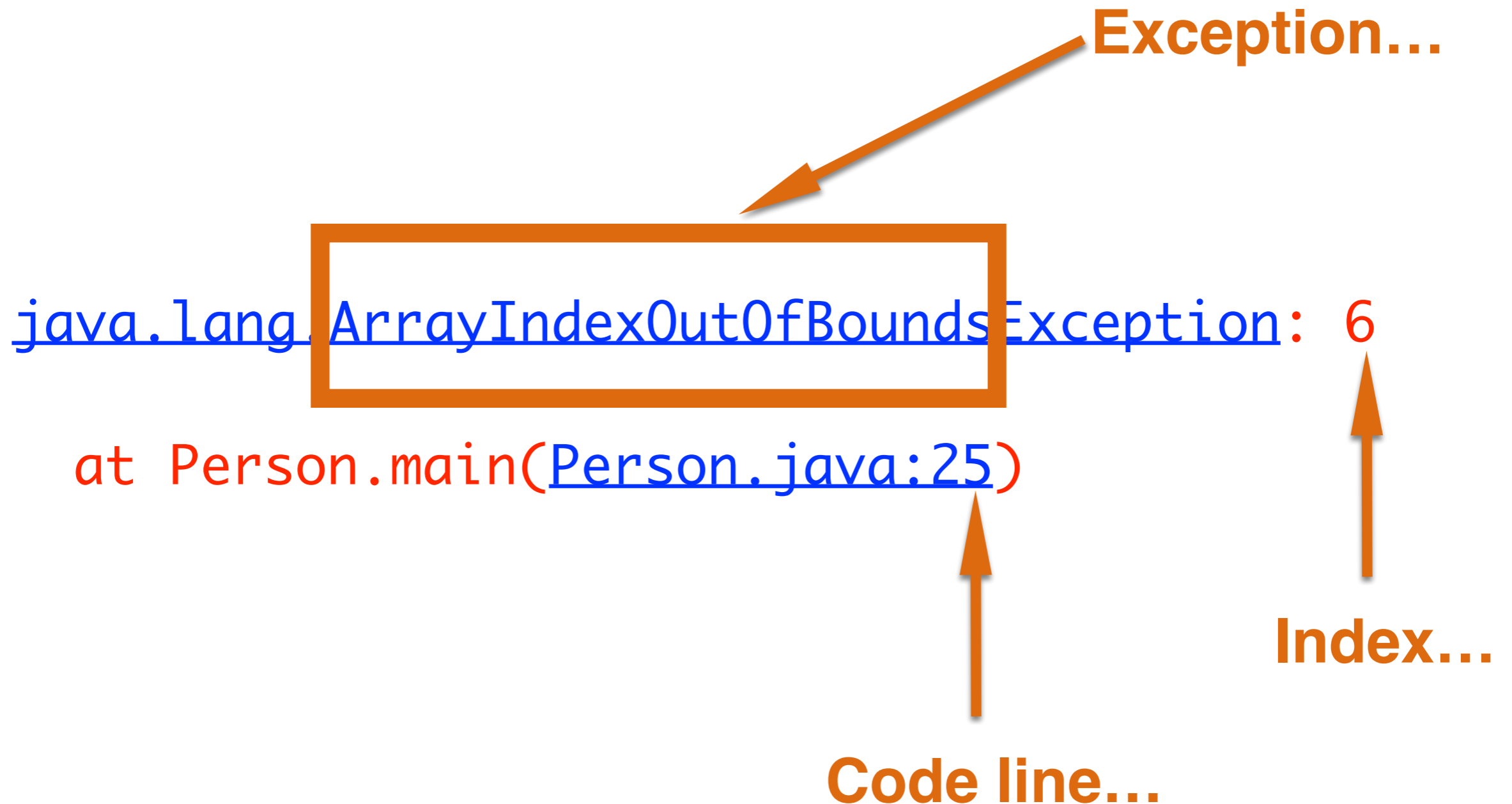
[java.lang.ArrayIndexOutOfBoundsException](#): 6

at Person.main([Person.java:25](#))



CRASH!

If your program accesses  
a cell outside the array...



# Examples

- A class for a **student**: creates an **array of 5** students ([link](#))

# Class Inheritance

# Class Inheritance

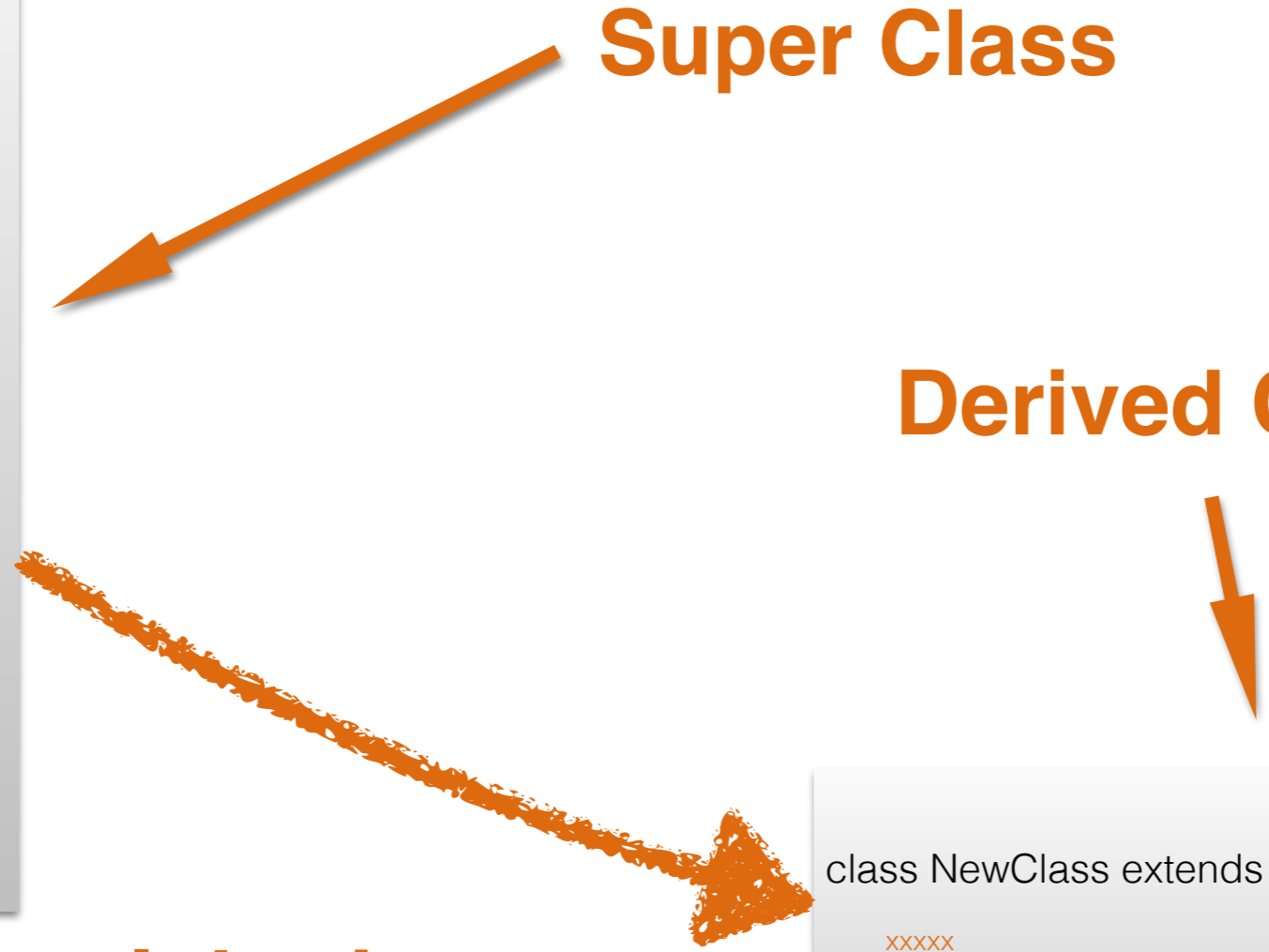
```
class BigClass {  
  XXXXXXXX X XXX  
  XXXXXX  
  XXXXXXXXXXXX  
  XXXXXXXXXXXX  
  XXXXXXXX  
  XXXXXXXXXXXX  
  XXXXXXXXXXXX  
  XXXXXX  
  XXXXXXXX  
  
  XXXXXX  
  XXXXXXXXXXXXXXXXXXXX  
  XXXXXXXX  
  
  XXXXXX  
  XXXXXX  
  XXXXXXXXXXXXXXXXXXXX  
  XXXXXXXX  
  XXXXXX  
  
  XXXXXXXXXXXXXXXXXXXX  
  XXXXXXXX  
  
  XXXXXX  
  XXXXXXXXXXXXXXXXXXXX  
  XXXXXXXX  
}
```

**Super Class**

**Derived Class**

**Inheritance**

```
class NewClass extends BigClass {  
  XXXXXX  
  XXXXXXXXXXXXXXXXXXXX  
  XXXXXXXX  
}
```



# Class Inheritance

```
class NewClass extends BigClass {
```

```
xxxxx
```

```
//class BigClass {
```

```
xxxxxxx x xxx
```

```
xxxxxx
```

```
xxxxxxxxxxx
```

```
xxxxxxxxxx
```

```
xxxxxxx
```

```
xxxxxxxxxxx
```

```
xxxxxxxxxxx
```

```
xxxxx
```

```
xxxxxxx
```

```
xxxxx
```

```
xxxxxxxxxxxxxxxxxxxx
```

```
xxxxxxx
```

```
xxxxxx
```

```
xxxxx
```

```
xxxxxxxxxxxxxxxxxxxx
```

```
xxxxxxx
```

```
xxxxx
```

```
xxxxxxxxxxxxxxxxxxxx
```

```
xxxxxxx
```

```
xxxxx
```

```
xxxxxxxxxxxxxxxxxxxx
```

```
xxxxxxx
```

```
// }
```

```
xxxxxxxxxxxxxxxxxxxx
```

```
xxxxxxx
```

```
}
```

# Examples

- A class for an Animal (super class)
- A class for a Bird (inherited from Animal)
- A class for a Dog (inherited from Animal)



## Animal

name  
age  
isVaccinated  
isTattooed  
Animal()  
displayBasicInfo()

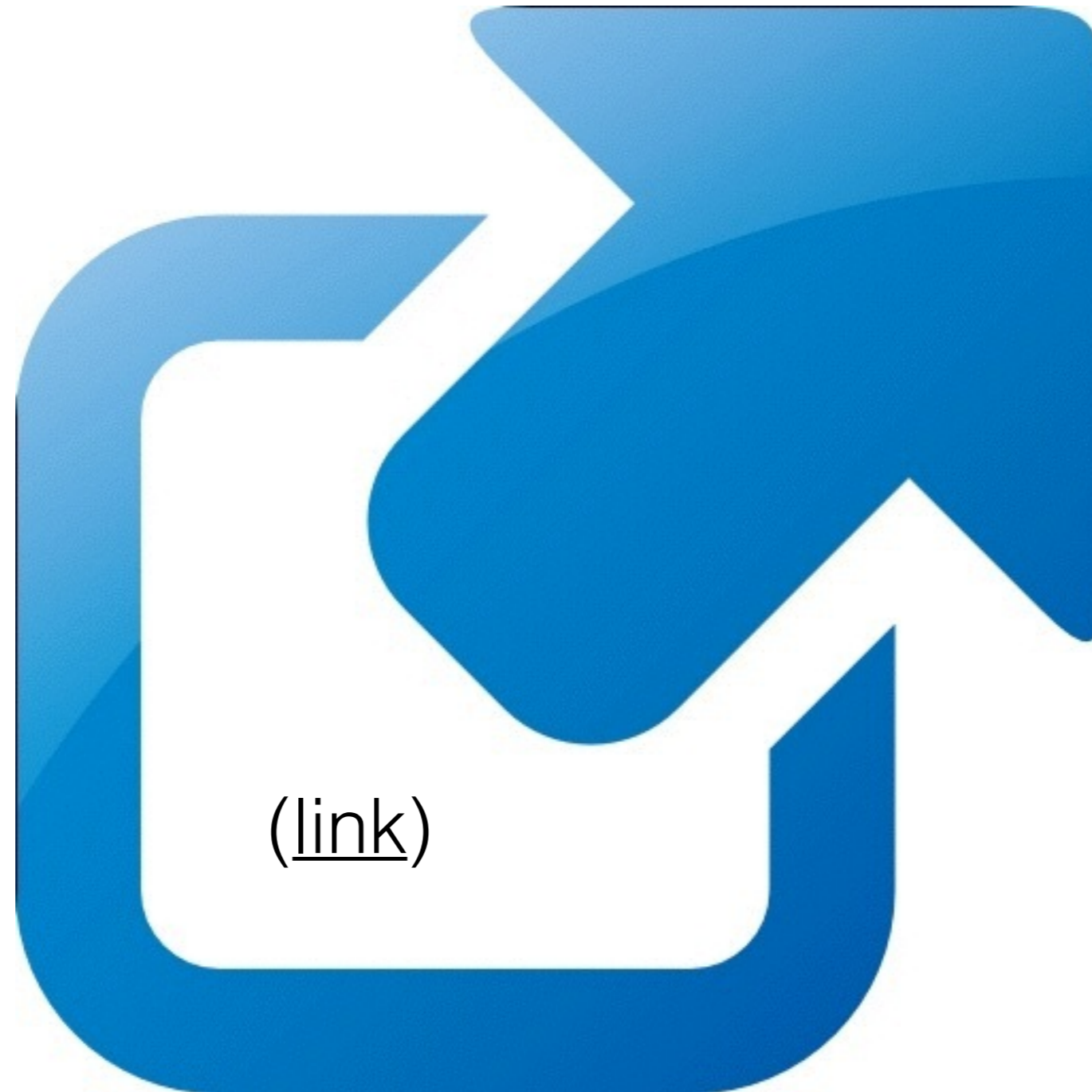
## Dog<—Animal

*name*  
*age*  
*isVaccinated*  
*isTattooed*  
legs[]  
*Animal()*  
setLegs()  
displayBasicInfo()

## Bird<—Animal

*name*  
*age*  
*isVaccinated*  
*isTattooed*  
wingSpan  
*Animal()*  
displayBasicInfo()

# Java Code for Examples



Beowulf2

```

[beowulf2]
[14:18:18] ~/public_html/classes/212$: ls -l *.java
-rw-rw-rw- 1 dthiebaut users 918 Sep 10 14:15 Animal.java
-rw-rw-rw- 1 dthiebaut users 817 Sep 10 14:15 Bird.java
-rw-rw-rw- 1 dthiebaut users 1376 Sep 10 14:15 Dog.java
-rw-rw-rw- 1 dthiebaut users 154 Sep 3 10:26 HelloWorld.java
[beowulf2]
[14:18:22] ~/public_html/classes/212$: javac Animal.java Dog.java Bird.java
[beowulf2]
[14:18:30] ~/public_html/classes/212$: java Animal
Max (3), tattooed, not vaccinated
[beowulf2]
[14:18:34] ~/public_html/classes/212$: java Bird
Toto (23), tattooed, vaccinated, wing-span: 34
Coco (17), not tattooed, vaccinated, wing-span: 29
[beowulf2]
[14:18:40] ~/public_html/classes/212$: java Dog
Rintintin (34), not tattooed, vaccinated, 2 bad leg(s)
Rex (4), tattooed, vaccinated, 1 bad leg(s)
[beowulf2]
[14:18:43] ~/public_html/classes/212$:

```