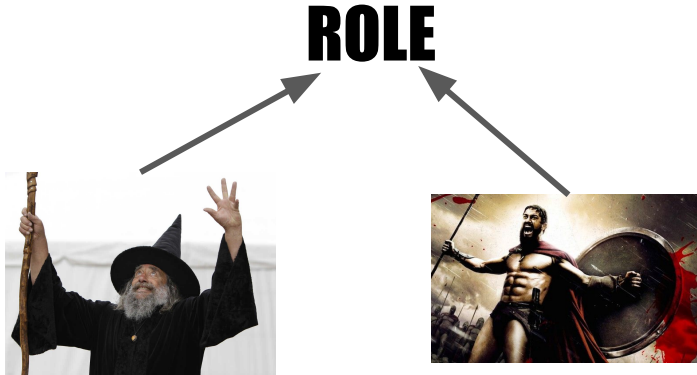# Inheritance

Mr. Poole
Java

Let's talk about *Interfaces* first.

Interfaces help us create one object type for multiple different objects.

- Example: we can make a Role that is either a Wizard or Warrior

This is great **BUT** we can still have code that is duplicated.
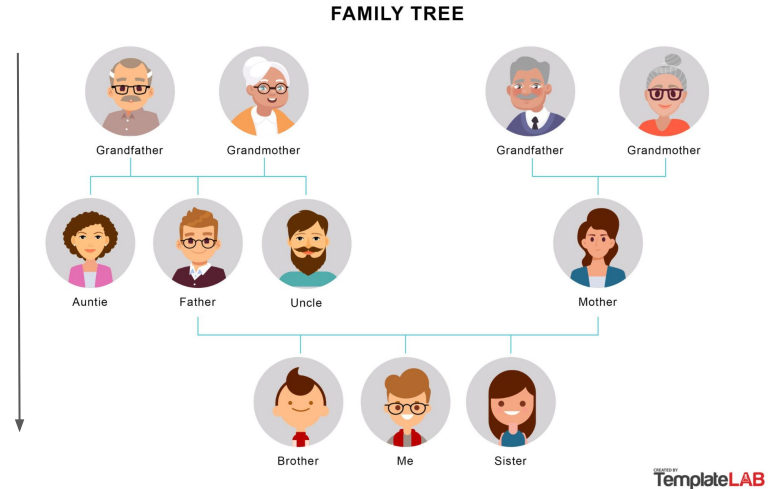
ROLE

Now let's introduce Inheritance

Inheritance allows us to **pass methods and values down** from the inherited class.

Passing information down helps **reduce the amount of code**!

It's just like family inheritance.

Parents pass traits down to their children.

Let's go over an example.

FAMILY TREE

Inheritance Example

Let's imagine we have these two classes.

Compare them!

What's different and what's the same?

Greyhound Class
- String breed = "Greyhound";
- void bark();
- boolean isFast();

Corgi Class
- String breed = "Corgi";
- void bark();
- boolean hasSmallLegs();

Inheritance Example

First off, they're **both dogs**!

They have different breeds and different methods.

They have the **same bark method** though!

Greyhound Class
- String breed = "Greyhound";
- void bark();
- boolean isFast();

Corgi Class
- String breed = "Corgi";
- void bark();
- boolean hasSmallLegs();

Inheritance Example

Since they're similar, let's make a general Dog class that can be inherited!

Dog class
- void bark();

Greyhound Class
- String breed = "Greyhound";
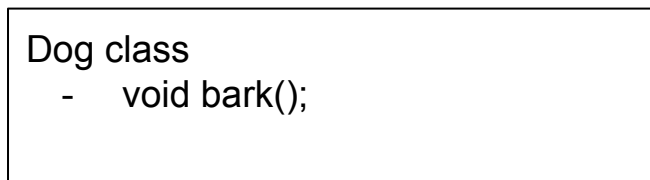- void bark();
- boolean isFast();

Corgi Class
- String breed = "Corgi";
- void bark();
- boolean hasSmallLegs();

Inheritance Example

Now to inherit the Dog's traits, we must say **extends** in our **subclasses**.

Dog class
- void bark();

Greyhound Class **extends Dog**
- String breed = "Greyhound";
- ~~void bark();~~
- boolean isFast();

Corgi Class **extends Dog**
- String breed = "Corgi";
- ~~void bark();~~
- boolean hasSmallLegs();

We can now delete the bark method from both classes since it's passed down!

Inheritance Example

These are our new classes! We've written the bark method one time less but all it adds up!

Dog class
- void bark();

bark() is passed down to the subclasses.

Greyhound Class **extends Dog**
- String breed = "Greyhound";
- boolean isFast();

Corgi Class **extends Dog**
- String breed = "Corgi";
- boolean hasSmallLegs();
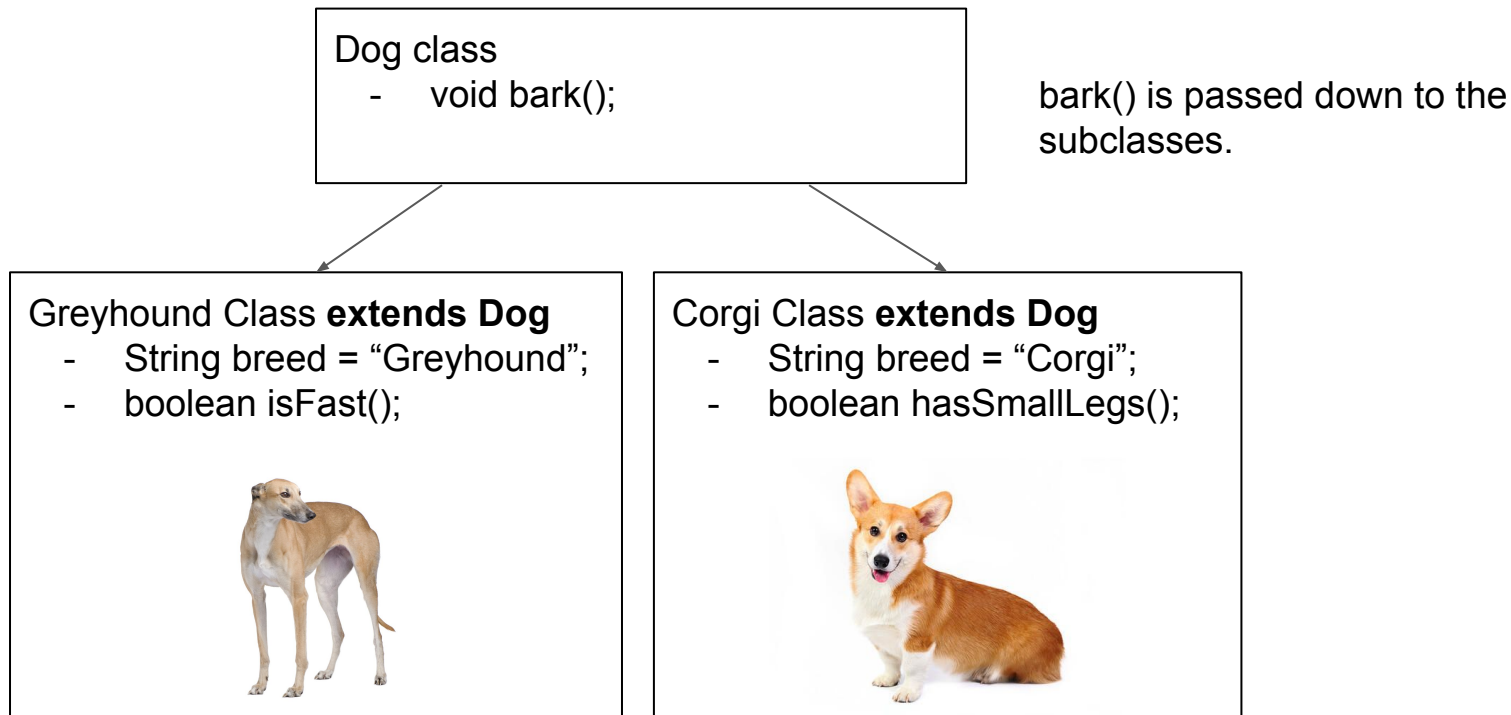
Now let's look at this in code.

Dog class
- void bark();

Greyhound Class **extends Dog**
- String breed = "Greyhound";
- boolean isFast();

Corgi Class **extends Dog**
- String breed = "Corgi";
- boolean hasSmallLegs();
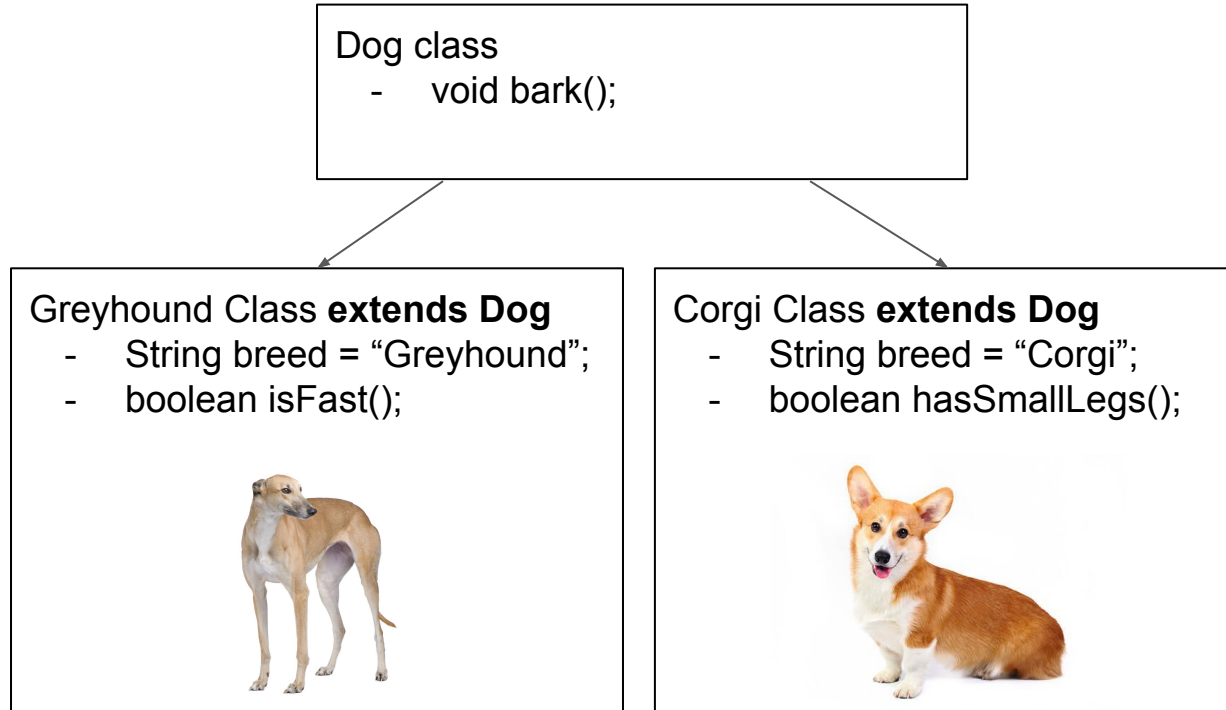
Inheritance - Corgi Class

As interfaces uses *implements*
Inheritance uses **extends**

This Corgi class **extends** the
Dog class.

This means that ANYTHING
that's defined in the Dog class
can be used by the Corgi class

```java
public class Corgi extends Dog{
    String breed;
    String name;

    public Corgi() {
        breed = "Corgi";
        name = "Joey";
    }
}
```
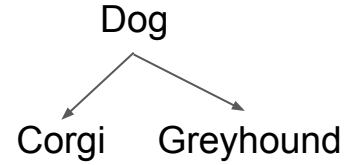
Inheritance - Dog Class

The Dog class is pretty generic.

It's just a class.
It has constructors and methods.
Unlike interfaces,
    this class expands the methods.

```java
public class Dog{
    String name;

    public Dog(){
        name = "Doggo";
    }


    public void bark(){
        System.out.println("Bark!");
    }
}
```

Inheritance - In combination

Dog

Corgi   Greyhound

Now, we can see that Corgi doesn't have a **bark()** method.

BUT since Dog does, we can use it!

```java
public class Corgi extends Dog{
    String breed;
    String name;

    public Corgi() {
        breed = "Corgi";
        name = "Joey";
    }
}
```

```java
public class Dog{
    String name;

    public Dog(){
        name = "Doggo";
    }

    public void bark(){
        System.out.println("Bark!");
    }
}
```

Inheritance - In combination

Given the previous slide, what does this code do?

```
Corgi a = new Corgi();
a.bark();
```

Inheritance - In combination

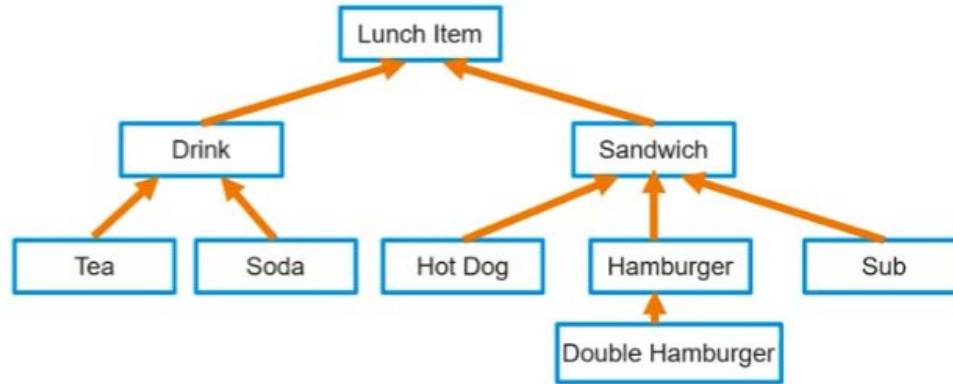Given the previous slide, what does this code do?

```
Corgi a = new Corgi();
a.bark();
```

It outputs "Bark!".
Even though Corgi doesn't have a bark method. It uses the Dog bark();

```
Bark!
```

Let's practice. Which are valid?

- public class Hamburger extends Sandwich
- public class Tea extends Drink
- public class Drink extends LunchItem
- public class Sandwich extends HotDog
- public class DoubleHamburger extends Sandwich
- public class Soda extends LunchItem
- public class DoubleHamburger extends Hamburger

Let's practice. Which are valid?
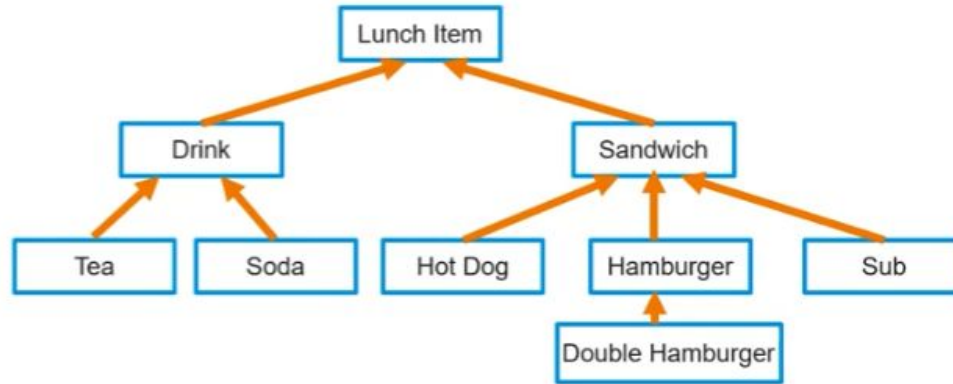
- public class Hamburger extends Sandwich ✔
- public class Tea extends Drink ✔
- public class Drink extends LunchItem ✔
- public class Sandwich extends HotDog ✘
- public class DoubleHamburger extends Sandwich ✘
- public class Soda extends LunchItem ✘
- public class DoubleHamburger extends Hamburger ✔

That's about it for Inheritance!
Methods and values can be passed down
from inherited classes.

Lab: Inheritance

- Look at the code given in basecode.
- Become familiar with the inheritance structure.
- Review Classes & Methods