

Why Bother With Inheritance?

Declared Type and Actual Type

```
DeclaredType varName = new ActualType(...);
```

```
Employee headChef = new Chef("Julia Child");
```

Declared Type: Employee

Actual Type: Chef

Can call methods that makes sense for EVERY Employee
If Chef overrides a method, uses the Chef version

```
Chef headChef = new Chef("Julia Child");
```

Declared Type: Chef

Actual Type: Chef

Can call methods that makes sense for EVERY Chef
If Chef overrides a method, uses the Chef version

Inheritance and Method Calls

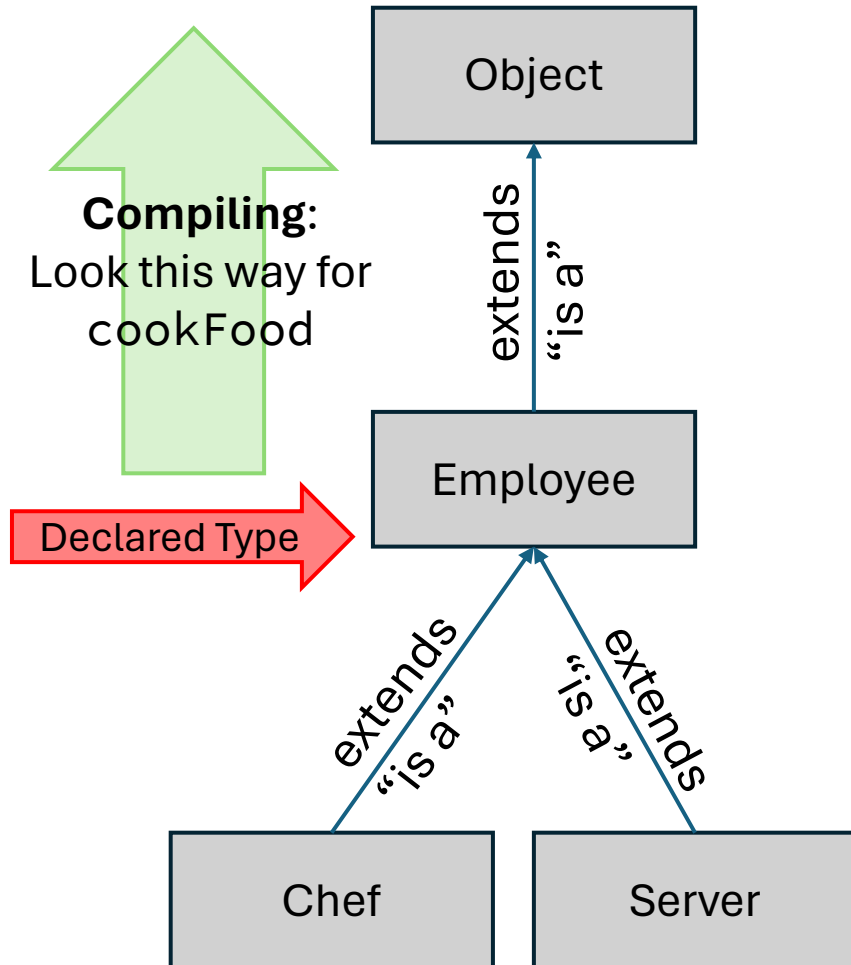
```
Employee headChef = new Chef("Julia Child");  
headChef.cookFood("potatoes");
```

When compiling:

Can we *guarantee* that the method exists for the **declared type**?

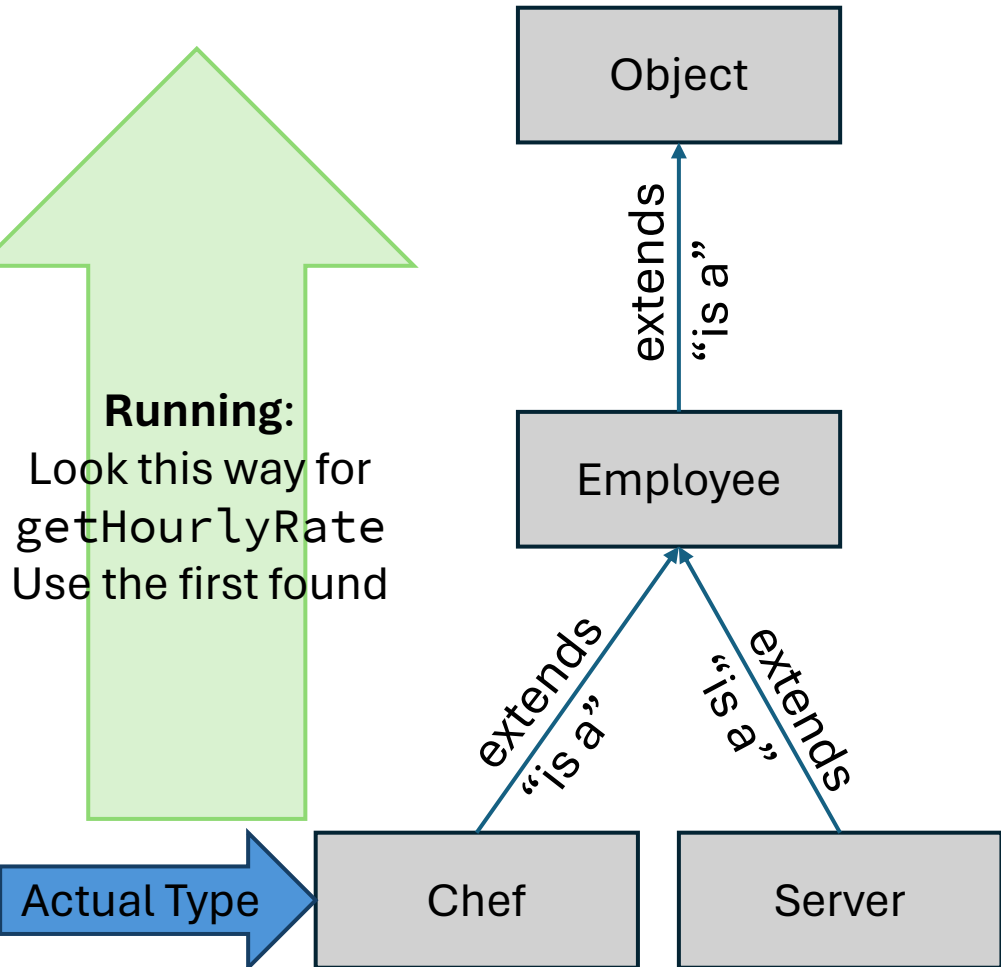
Does the **declared type** or one of its super classes contain a method of that name?

If not... Compile Error!



Overrides and Method Calls

```
Employee headChef = new Chef("Julia Child");  
headChef.getHourlyRate();
```



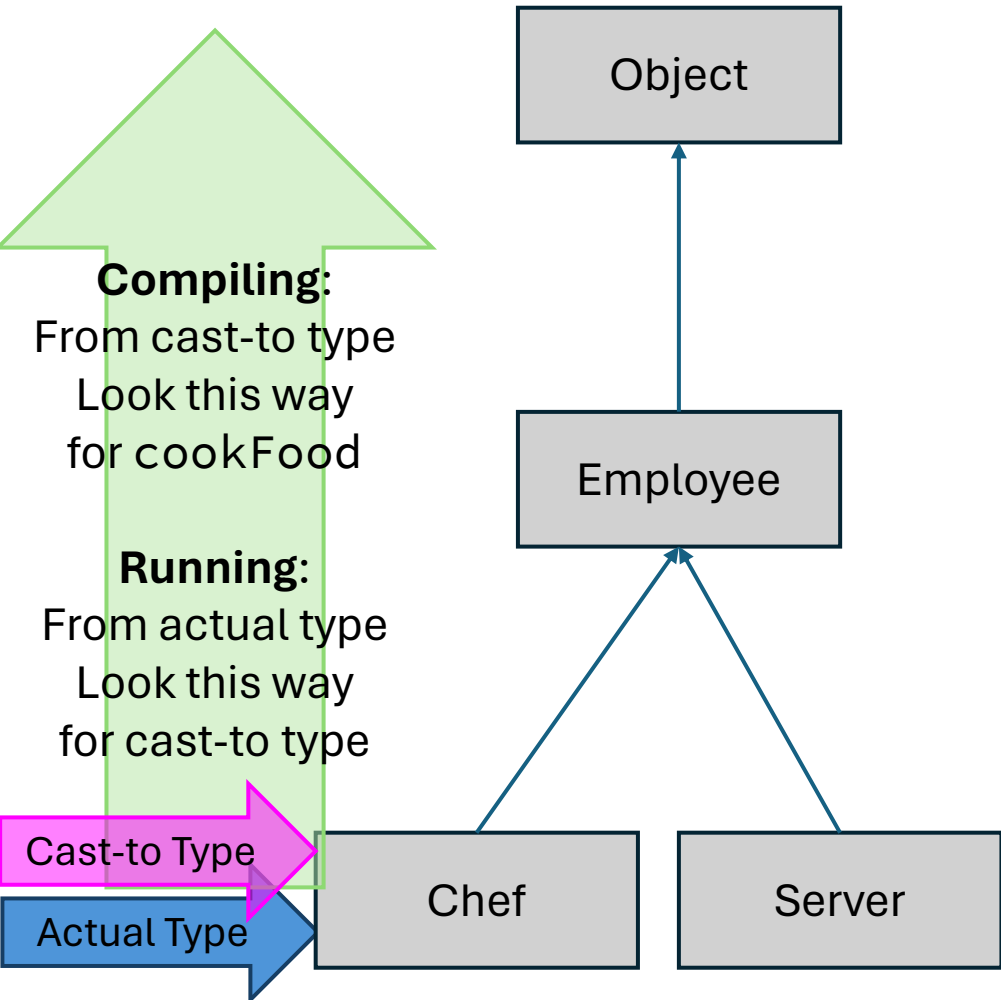
When running:

Use the *most specific* version of the method call starting from the **actual type**.

Start from the **actual type**, then go “up” to super classes until you find the method.
Run that first-discovered version.

Casting and Method Calls

```
Employee headChef = new Chef("Julia Child");  
((Chef) headChef).cookFood("potatoes");
```



When compiling:

Can we *guarantee* that the method exists for the **Cast-to type**?

Does the **Cast-to type** or one of its super classes contain a method of that name?

If not... Compile Error!

When Running:

Check that the **Cast-to Type** is either the **Actual Type**, or one of its super classes

```
public class Employee
    public int getHours()
    public int getVacationDays()
    public String toString()
```

```
public class Astronaut extends Employee
    public void takeoff()
    public String toString()
    public int getHours()
```

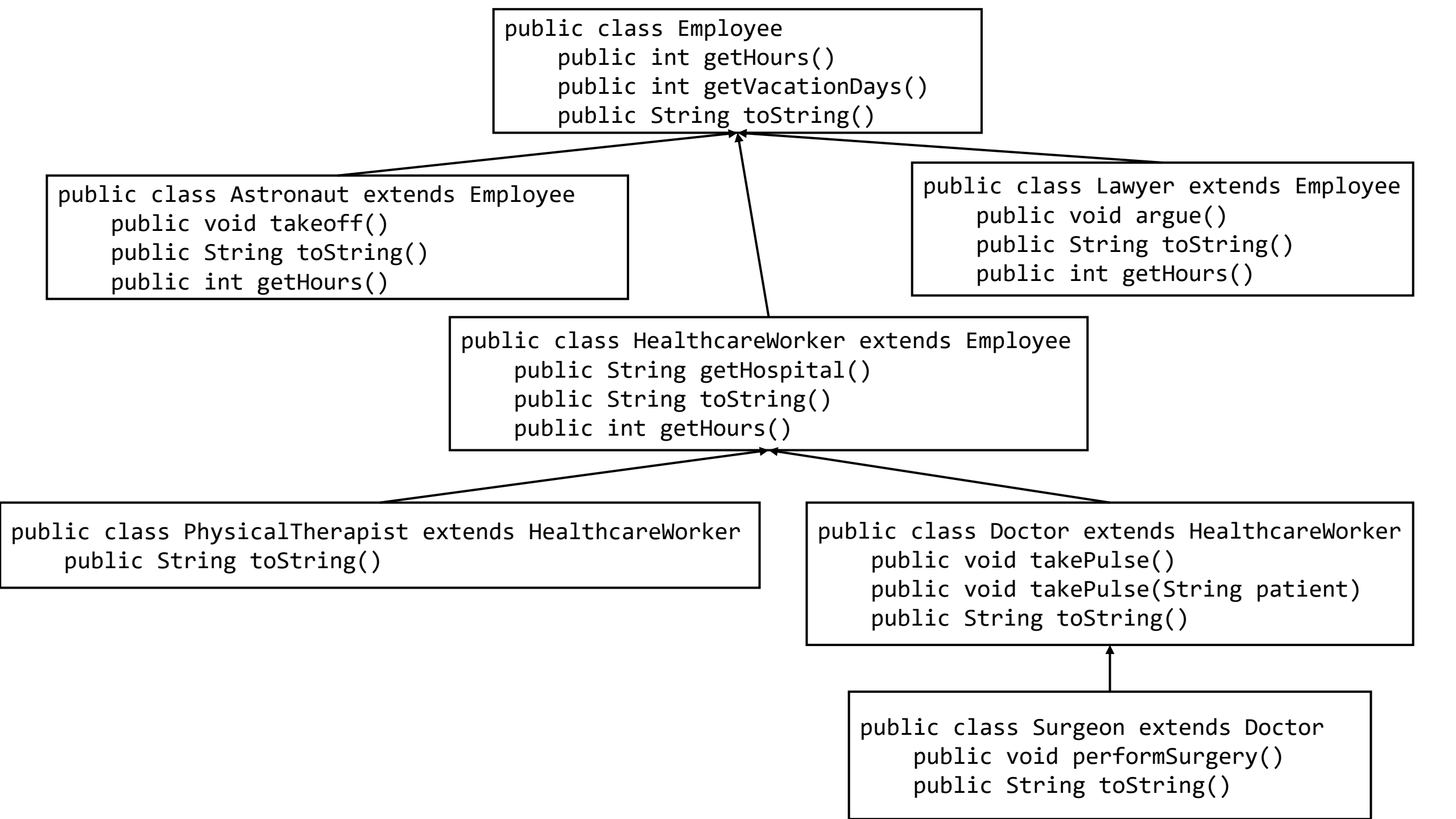
```
public class Lawyer extends Employee
    public void argue()
    public String toString()
    public int getHours()
```

```
public class HealthcareWorker extends Employee
    public String getHospital()
    public String toString()
    public int getHours()
```

```
public class PhysicalTherapist extends HealthcareWorker
    public String toString()
```

```
public class Doctor extends HealthcareWorker
    public void takePulse()
    public void takePulse(String patient)
    public String toString()
```

```
public class Surgeon extends Doctor
    public void performSurgery()
    public String toString()
```



```
public class Employee
    public int getHours()
    public int getVacationDays()
    public String toString()
```

```
public class Astronaut extends Employee
    public void takeoff()
    public String toString()
    public int getHours()
```

```
public class Lawyer extends Employee
    public void argue()
    public String toString()
    public int getHours()
```

```
public class HealthcareWorker extends Employee
    public String getHospital()
    public String toString()
    public int getHours()
```

```
public class PhysicalTherapist extends HealthcareWorker
    public String toString()
```

```
public class Doctor extends HealthcareWorker
    public void takePulse()
    public void takePulse(String patient)
    public String toString()
```

```
public class Surgeon extends Doctor
    public void performSurgery()
    public String toString()
```

