

New Idea –
Extending Concrete
Class

Mr. Neat
Java

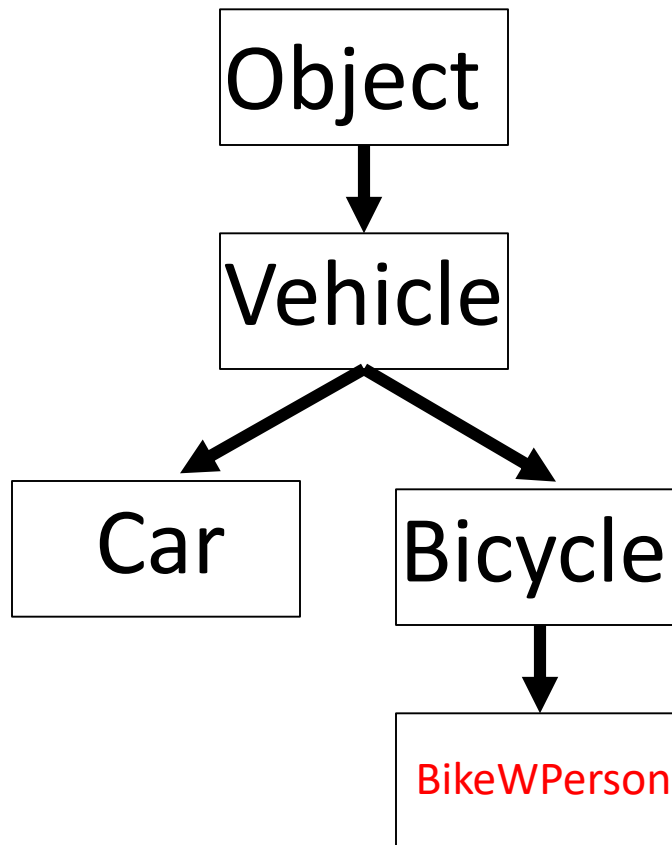
Big Idea in Java...

Inheritance

- Mechanism to make a group of different objects that have something in common
- Mechanism to make a more specialized class from a more general class (eg, Car with person in it)
- The Object class is the mother of all classes
- Family trees help to understand

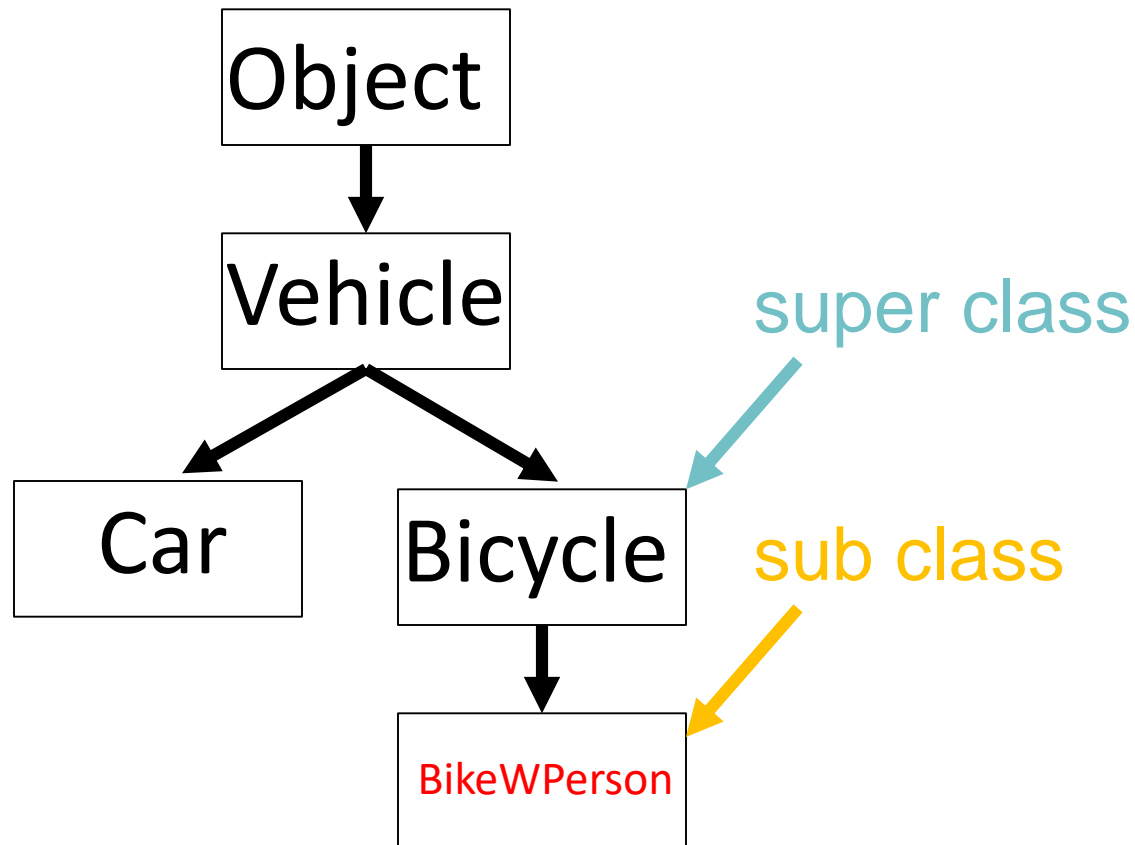
Big Idea in Java...

Inheritance – Extending a Concrete class



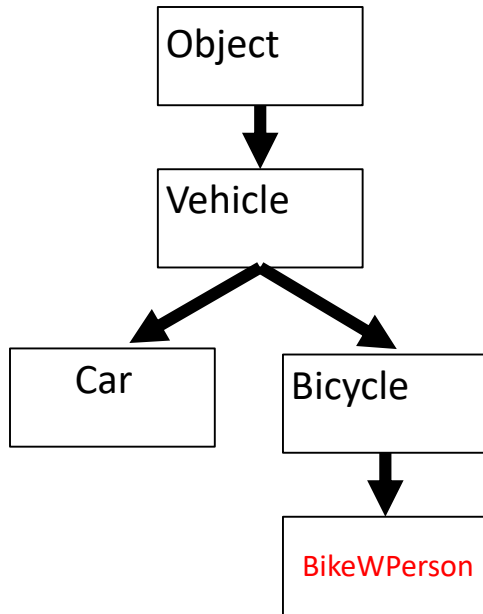
Big Idea in Java...

Inheritance – Extending a Concrete class



Big Idea in Java...

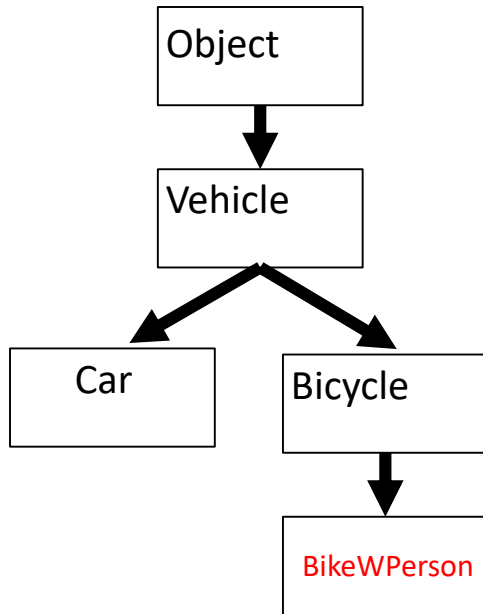
Inheritance – Extending a Concrete class



- Make a new file called **BikeWPerson**
- Extend Bicycle class
- Call super classes constructors
- Only include in the code:
 - private fields needed for person
 - Methods needed for the person
- Everything else is inherited from the super class!

Big Idea in Java...

Inheritance – Extending a Concrete class



- Make a new file called BikeWPerson
- Extend Bicycle class
- Call super classes constructors
- Only include in the code:
 - private fields needed for person
 - Methods needed for the person
- Everything else is inherited from the super class!

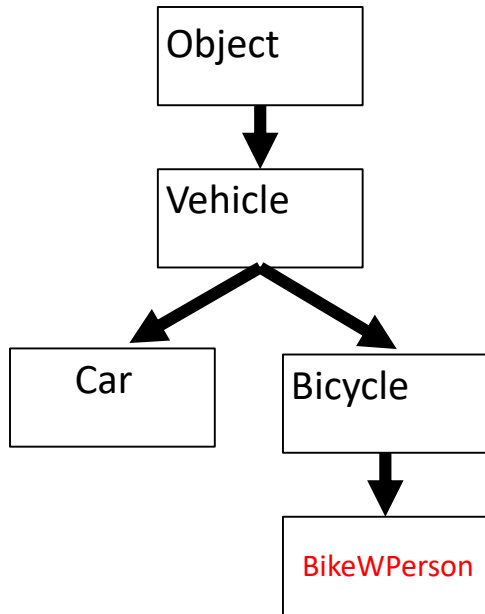
```
public class BikeWPerson extends Bicycle
{

}
```

Big Idea in Java...

Inheritance – Extending a Concrete class

- Make a new file called BikeWPerson
- Extend Bicycle class
- **Call super classes constructors**
- Only include in the code:
 - private fields needed for person
 - Methods needed for the person
- Everything else is inherited from the super class!



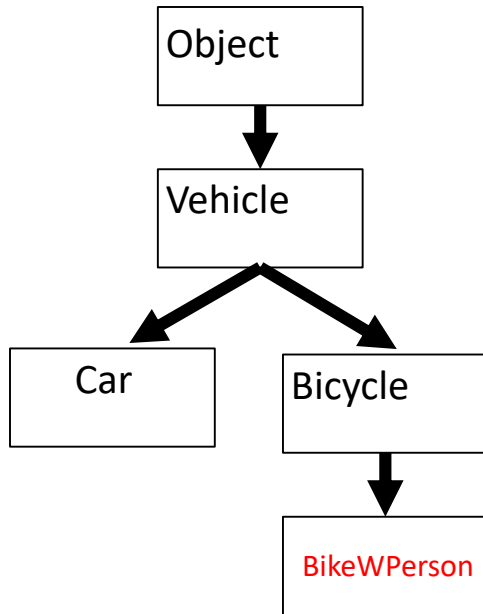
```
public BikeWPerson(int x, int y)  
{
```

```
    super(x,y);  
    // stuff unique to person
```

Calls super classes
constructor with same
parameter set

Big Idea in Java...

Inheritance – Extending a Concrete class



- Make a new file called BikeWPerson
- Extend Bicycle class
- Call super classes constructors
- Only include in the code:
 - **private fields needed for person**
 - Methods needed for the person
- Everything else is inherited from the super class!

```
public class BikeWPerson extends Bicycle  
{
```

```
// add fields unique to person
```

```
//all the rest of the fields are already defined in the super classes.
```

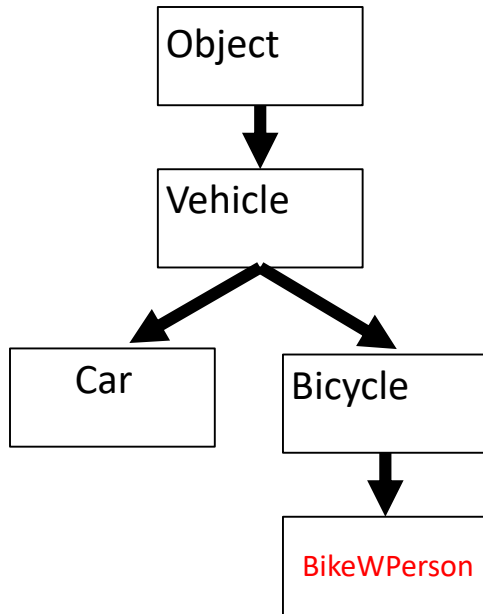
```
// DON'T REDEFINE THEM AGAIN!
```

```
}
```


Big Idea in Java...

Inheritance – Extending a Concrete class

- Make a new file called BikeWPerson
- Extend Bicycle class
- Call super classes constructors
- Only include in the code:
 - private fields needed for person
 - **Methods needed for the person**
- Everything else is inherited from the super class!



```
public void translate(double ex, double why)
{
    super.translate(ex,why);
    head.translate(ex,why);
    body.translate(ex,why);
}
```

Calls super classes
method

moves the person parts

Big Idea in Java...

Inheritance – Extending a Concrete class

NOTE:

Private instance variables of super classes are NOT inherited by sub classes. So if you need access to a super class private variable, that super class needs to have an accessor method for that variable. This issue may require you to go back and edit your super class. In the future, load your super class with accessor methods in anticipation of others extending your class.

Lab

- Make a BikeWPerson class
- Extend the provided NeatoBicycle class
- Inherit the NeatoBicycle toString method and augment it to include a person
- Test your code by constructing one object from the new class and recycling it. Also, SOP a BikeWPerson object

Hints

- NeatoBicycle constructors:
 - `public NeatoBicycle(double x, double y)`
 - `public NeatoBicycle(double x, double y, String label)`
 - `public NeatoBicycle(double x, double y, String label, Color c)`
 - `public NeatoBicycle(double x, double y, String label, Color c, int step)`
- NeatoBicycle methods:
 - `public void draw()`
 - `public void translate(double ex, double why)`
 - `public void driveRandom()`
 - `public int getX()`
 - `public int getY()`
 - `public int getStep()`
 - `public void setStep(int st)`
 - `public int getHeight()`
 - `public int getWidth()`
 - `public Rectangle getBoundingBox()`
 - `public String getName()`
 - `public String toString()`

Hints (continued)

- Place the NeatoBicycle.class file in your pkg folder
- This file is located in this folder (notes)