

Blueprints and Instances

Objects: instances of classes



Is the cookie cutter a cookie?

PAUSE AND TRY

Talk with a partner about one cookie cutter and a tray of cookies.

Is the cookie cutter a cookie?

How many cookies can one cutter make?

Do all the cookies have the same shape?

Can two cookies still be different? How?

By the end of class you can...

- 1 Explain how a class and an object are related, using String.
- 2 Tell what every object of a class shares from what each one has on its own.
- 3 Read a class hierarchy and list what a subclass gets from its superclass.
- 4 Explain why every class in Java is a subclass of Object.
- 5 Explain that a reference variable holds a reference, and trace copies and reassignments.

A large white square with rounded corners, tilted slightly to the right, containing a large teal number '1'.

PART 1

Classes and objects

A class is the blueprint. An object is one instance.

Class

THE BLUEPRINT

Defines what every object of its kind knows (attributes) and can do (behaviors).

String

Object (instance)

ONE SPECIFIC THING

Made from a class. Has the class's attributes, with its own values.

"Kai" "Marisol" "Jo Ann"



Same kinds of attributes and the same behaviors for every object. **Its own values** for each one.

Three objects, one class

```
String first = "Kai";
String second = "Marisol";
String third = "Jo Ann";
System.out.println(first.length());
System.out.println(second.length());
System.out.println(third.length());
```

OUTPUT

3
7
6

In the code	Class or object?	length()
String	Class: the plan	(not asked)
"Kai"	Object	3
"Marisol"	Object	7
"Jo Ann"	Object	6 (the space counts)

"String is an object"



String is the class. **"Kai"** is an object, one instance of String.

THE WHICH-ONE TEST

Which string? The one that says "Kai". You can ask "which one?" about objects.

Which String class? That question makes no sense. There is only one definition.

SAME CLASS IS NOT THE SAME OBJECT

"Kai" and "Marisol" are both String objects.

Same behaviors, same kinds of attributes.
Different characters, different lengths.

Class or object?

PAUSE AND TRY

Label each one: class or object?

Scanner

"pizza"

String

what new Scanner(System.in) creates

" " (no characters at all)

Test:

Can you ask "which one?"
Then it is an object.

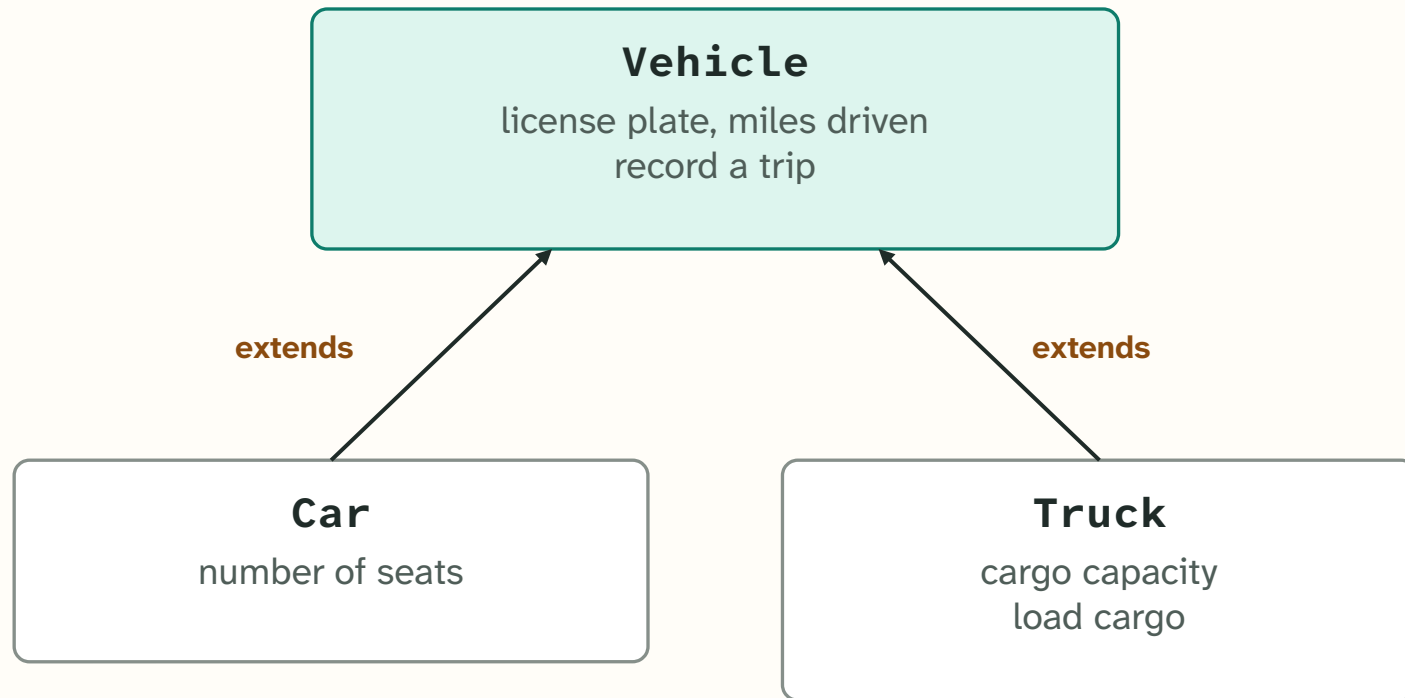
A large, white, rounded square containing a bold teal number '2', indicating the second part of a series.

2

PART 2

Class hierarchies

A superclass holds what subclasses share



Superclass

Holds the shared attributes and behaviors.

Subclass

Extends a superclass. Gets what it has, adds its own.

Inheritance

The relationship from a subclass to its superclass.

A truck is a vehicle. A car is a vehicle.

Everything a Truck object has

Step	Look in	Attributes	Behaviors
1	Truck	cargo capacity	load cargo
2	Vehicle	license plate, miles driven	record a trip
Total	A Truck object	3 attributes	2 behaviors



Change it once. If the company adds a fuel level to Vehicle only, every Car object and every Truck object now has a fuel level, and nobody edited Car or Truck.

Inheritance only climbs up



Start at the class you are asked about and move **up**. Never down, never sideways.

DOWN: NO

Does a Vehicle object have cargo capacity?

No. Cargo capacity is written in Truck, a subclass. A superclass gets nothing from its subclasses.

SIDEWAYS: NO

Does a Car object have cargo capacity?

No. Car and Truck are siblings. They share only what is in Vehicle.

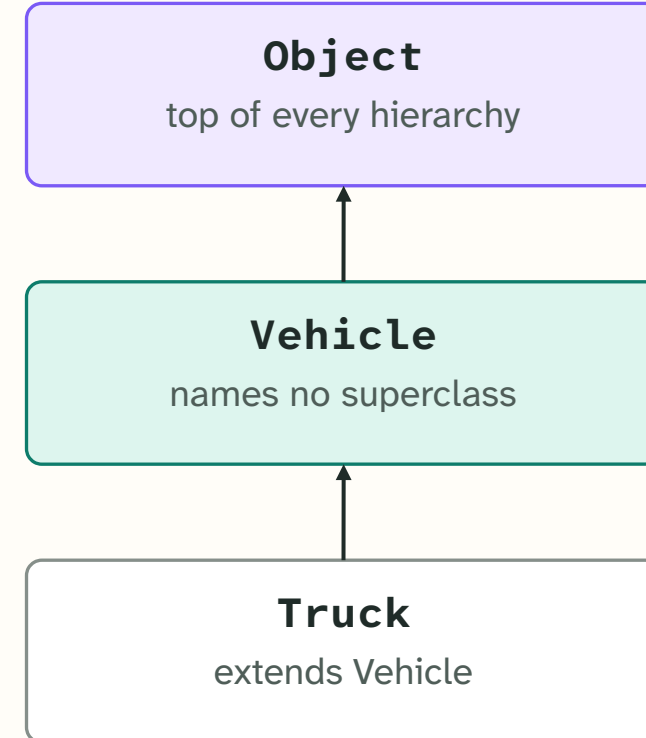
Every class is a subclass of Object

FROM THE STRING DOCUMENTATION

```
java.lang.Object  
java.lang.String
```

FROM THE SCANNER DOCUMENTATION

```
java.lang.Object  
java.util.Scanner
```

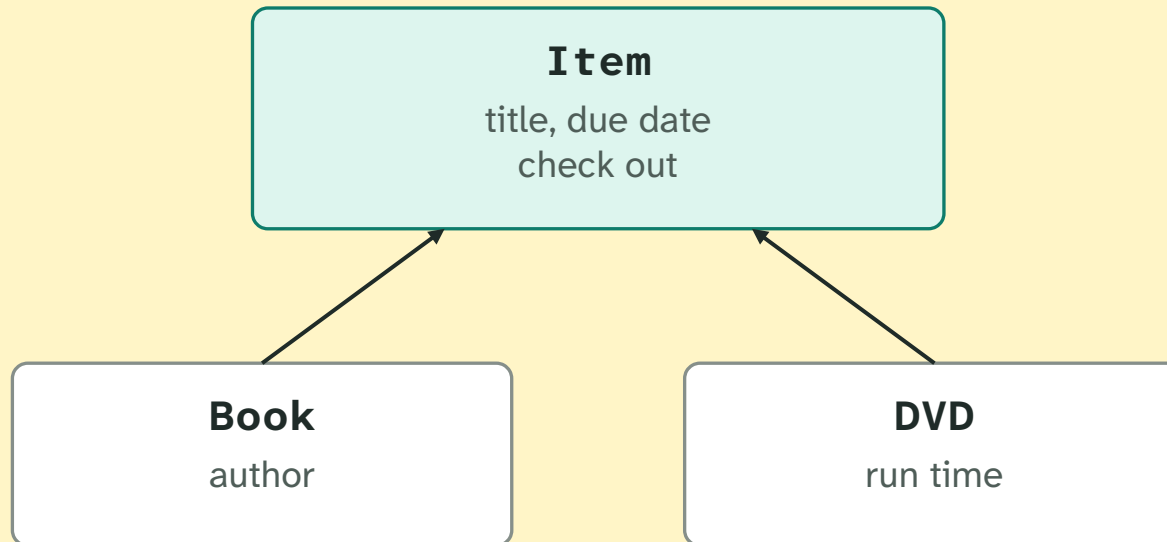


Object is one class. An **object** is any instance of any class.

Climb the library's tree

PAUSE AND TRY

Item names no superclass. Book and DVD extend Item.



1. List everything a DVD object has.
2. Does a Book object have a run time? Why?
3. What is the superclass of Item?

A large, white, rounded square containing the number '3' in a bold, teal, sans-serif font.

PART 3

Reference variables

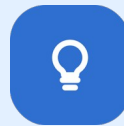
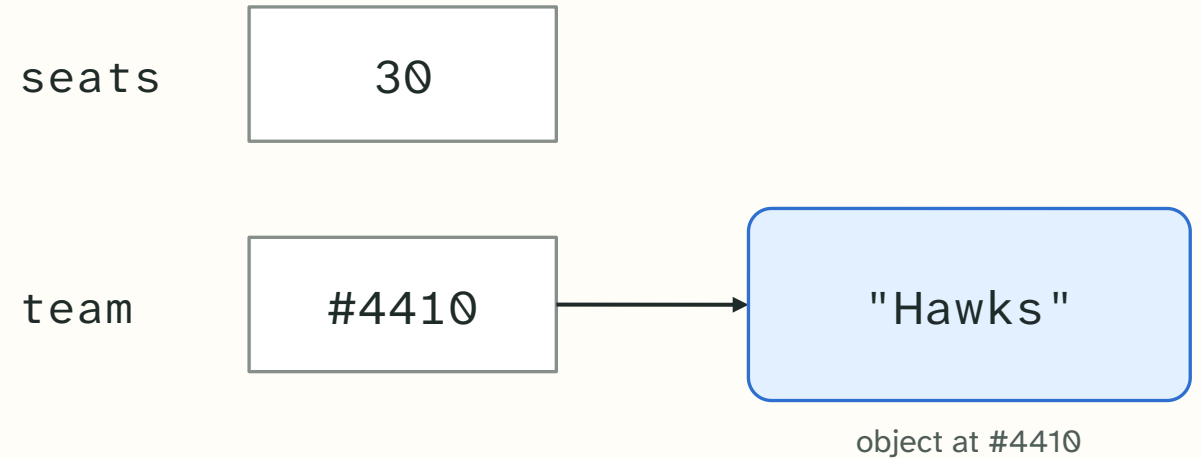
A reference variable holds a reference

```
int seats = 30;  
String team = "Hawks";  
System.out.println(seats);  
System.out.println(team);
```

OUTPUT

```
30  
Hawks
```

MEMORY (ADDRESS MADE UP)



A primitive box holds the **value**. A reference box holds a **reference**: where to find the object.

Copy a reference, then reassign

```
1 String home = "Reno";
2 String trip = home;
3 home = "Tulsa";
4 System.out.println(trip);
5 System.out.println(home);
```

OUTPUT

```
Reno
Tulsa
```

After line	home refers to	trip refers to
1	"Reno"	(not declared)
2	"Reno"	"Reno" (same)
3	"Tulsa"	"Reno"

Copying a reference copies the reference. No new object is made.

"Change one, and the other changes too"



Each variable has its **own box**. Reassigning one variable never changes what another variable holds.

NOTE 1: HOME

Had the address of "Reno".
Now has the address of
"Tulsa".

NOTE 2: TRIP

Got a copy of the "Reno"
address.
Still has it.

THE OBJECTS

"Reno" is still there.
"Tulsa" is a different object.

Track every box

PAUSE AND TRY

Write the exact output. Trace what each variable refers to after each line.

```
1 String x = "cat";
2 String y = "dog";
3 String z = x;
4 x = y;
5 y = "emu";
6 System.out.println(x);
7 System.out.println(y);
8 System.out.println(z);
```

After line	x	y	z
3			
4			
5			

Words to know

Class

The blueprint of the attributes and behaviors of a kind of object.

Object (instance)

One specific thing made from a class, with its own attribute values.

Superclass / subclass

Holds what related classes share / extends a superclass and draws on it.

Inheritance

The relationship from a subclass up to its superclass.

Object (the class)

The class at the top of every hierarchy. Every class is a subclass of it.

Object reference

What a reference variable holds: think of it as the object's memory address.

One blueprint, many objects



Explain to your partner how `String` and `"Kai"` are related, and what the variable in `String name = "Kai";` holds.

SENTENCE STARTER

`String` is the _____, `"Kai"` is one _____ of it, and `name` holds a _____ to that object.

Next: the Blueprints and Instances puzzle worksheet.