

Inside the Class

Anatomy of a class



What can you reach?

PAUSE AND TRY

A vending machine: what can customers use? What is locked inside?

CUSTOMERS CAN USE

LOCKED INSIDE

Then: why would the maker lock some parts away?

By the end of class you can...

- 1 Find the parts of a class: header, instance variables, constructor, methods.
- 2 Tell a constructor from a method.
- 3 Explain why each object has its own copy of each instance variable, and trace it.
- 4 Decide whether code can use a member, based on public and private.
- 5 Explain encapsulation and why instance variables are private.

A large, white, rounded square containing the number '1' in a bold, teal, sans-serif font.

1

PART 1

The parts of a class

A class, part by part

STOPWATCH.JAVA

```
public class Stopwatch {  
    private int seconds;  
    public Stopwatch() {  
        seconds = 0;  
    }  
    public void tick() {  
        seconds++;  
    }  
    public void printTime() {  
        System.out.println(seconds + " s");  
    }  
}
```

Class header (line 1)

public class Stopwatch: the keyword class and the name.

Instance variable (line 2)

Data each object keeps. Inside the class, outside every method.

Constructor (lines 3-5)

Same name as the class, no return type. new runs it.

Methods (lines 6-11)

Behaviors: tick adds a second, printTime shows it.

Constructor or method?

CONSTRUCTOR

```
public Stopwatch() {  
    seconds = 0;  
}
```

- Same name as the class
- No return type
- Runs when new builds an object

METHOD

```
public void tick() {  
    seconds++;  
}
```

- Its own name
- A return type, here void
- Runs when you call it on an object



Writing void in front of a constructor's name makes it a method. **new will never run it.**

A large white square with rounded corners, tilted slightly to the right, containing the number '2' in a teal font.

2

PART 2

Every object has its own copy

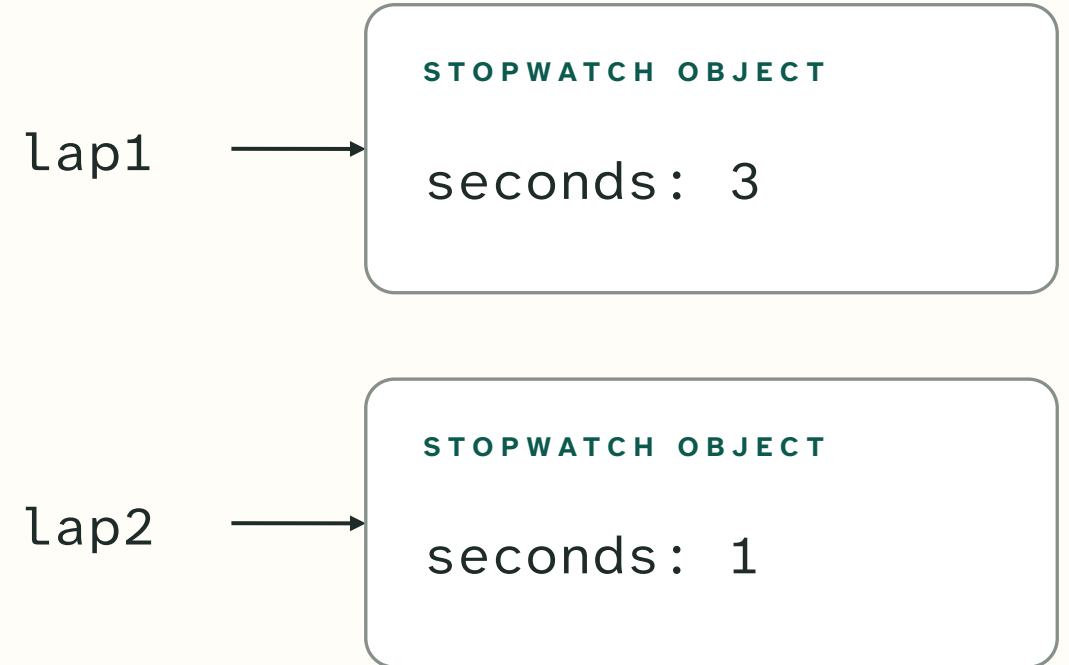
Two stopwatches, two seconds

MAIN, IN ANOTHER CLASS

```
Stopwatch lap1 = new Stopwatch();  
Stopwatch lap2 = new Stopwatch();  
lap1.tick();  
lap1.tick();  
lap1.tick();  
lap2.tick();  
lap1.printTime();  
lap2.printTime();
```

OUTPUT

```
3 s  
1 s
```



Two objects, two separate copies of seconds.

Whose cookies?

PAUSE AND TRY

Each new Jar starts with 10 cookies; eat() removes one. What prints?

MAIN

```
Jar a = new Jar();  
Jar b = new Jar();  
a.eat();  
b.eat();  
b.eat();  
a.printJar();  
b.printJar();
```

A METHOD OF JAR

```
public void printJar() {  
    System.out.println(cookies  
        + " cookies");  
}
```

Write both lines of output.



3

PART 3

public and private

Who may use it?



public: any class may use it. **private:** only code inside the class that declares it.

Where the code is	Use a public member?	Use a private member?
A constructor or method of the same class	Yes	Yes
Code in a different class, such as its main	Yes	No: it does not compile

Outside code hits the private door

MAIN, IN ANOTHER CLASS

```
Stopwatch lap1 = new Stopwatch();  
lap1.seconds = 99;
```

OUTPUT

```
Cheat.java:4: error: seconds has  
private access in Stopwatch
```

USE THE PUBLIC METHODS INSTEAD

```
Stopwatch lap1 = new Stopwatch();  
lap1.tick();  
lap1.printTime();
```

OUTPUT

```
1 s
```



Private does not mean unusable. It means **usable only from inside**. The public methods are the way in.

Same file is not "inside the class"



Only code between the class's **own braces** is inside. A `main` in another class is outside, even in the same file.

ONE FILE, TWO CLASSES

```
public class Race {  
    public static void main(String[] args) {  
        // outside Stopwatch  
    }  
}  
  
class Stopwatch {  
    private int seconds;  
    // constructors and methods here are inside  
}
```

Ask:

Which class's braces is this line between? If it is not the class that declared the private member, it cannot use it.

Which lines compile?

PAUSE AND TRY

Each line is in main, after `Robot r = new Robot();`

ROBOT (BODIES HIDDEN)

```
class Robot {  
    public String name;  
    private int battery;  
    public Robot() { ... }  
    public void charge() { ... }  
    private void beep() { ... }  
}
```

A `r.name = "Rex"; r.charge();`

B `r.battery = 100;`

C `r.charge();`

D `r.beep();`

4

PART 4

Encapsulation

Keep the data inside

IF SECONDS WERE PUBLIC

```
// with: public int seconds;  
Stopwatch lap1 = new Stopwatch();  
lap1.seconds = -30;  
lap1.printTime();
```

OUTPUT

-30 s



Encapsulation keeps a class's implementation details hidden inside it. Other classes use only its public parts, so they cannot put an object into a nonsense state.

The rule: mark every instance variable **private** unless the class's specification says it is public.

Who wears which keyword

Part	Keyword	Why
Class	<code>public</code>	In this course, every class; one public class per file, named like the file
Constructor	<code>public</code>	Other classes must be able to build objects with <code>new</code>
Instance variable	<code>private</code>	Encapsulation, unless the specification says <code>public</code>
Method other classes call	<code>public</code>	A service, like <code>tick()</code> or <code>printTime()</code>
Helper method	<code>private</code>	Used only by the class's own code

Words to know

Class header

public class Name: the first line of a class.

Instance variable

Data each object keeps; declared outside every method.

Constructor

Class name, no return type; new runs it to set starting values.

Access modifier

public or private: which code may use a member.

Encapsulation

Keeping a class's details hidden; others use its public parts.

Helper method

A private method that only the class's own code calls.

Lock it or leave it open?



Explain to your partner why a class makes its instance variables private but its constructor public.

SENTENCE STARTER

Instance variables are private so that _____. The constructor is public so that _____.

Next: the Inside the Class puzzle worksheet.