

Numbers in Envelopes

Wrapper classes: Integer and Double



Six plus seven is...?

PAUSE AND TRY

What does this print? How would you get 13 instead?

JOIN.JAVA

```
String a = "6";  
String b = "7";  
System.out.println(a + b);
```

Hint: look at the types of a and b.
Where have you seen numbers stored as text this unit?

By the end of class you can...

- 1 Explain how an Integer or Double object differs from an int or double.
- 2 Use Integer.MAX_VALUE and Integer.MIN_VALUE as safe starting values.
- 3 Predict where Java autoboxes a value and where it unboxes one.
- 4 Explain why an Integer or Double can never change.
- 5 Turn text into numbers with Integer.parseInt and Double.parseDouble.

A white, rounded square with a slight 3D effect, containing a large, bold, teal-colored number '1' in the center.

1

PART 1

A number inside an object

Primitive value or wrapper object?

```
int laps = 12;
```

laps

12

holds the number itself

```
Integer count = 12;
```

count

ref

Integer object

value: 12

holds a reference (or null)

Primitive	Wrapper class
int	Integer
double	Double



A **wrapper class** object holds one primitive value. **Integer** and **Double** are in the **java.lang** package, so no import is needed.

A start that any value beats

SHORTEST WAIT, IN MINUTES

```
int[] waits = {14, 6, 9, 11};
int shortest = Integer.MAX_VALUE;
for (int w : waits) {
    if (w < shortest) {
        shortest = w;
    }
}
System.out.println(shortest);
```

OUTPUT

6

w	w < shortest?	shortest after
(start)		2147483647
14	yes	14
6	yes	6
9	no	6
11	no	6

Minimum: start at **Integer.MAX_VALUE**. Maximum: start at **Integer.MIN_VALUE**.

Starting the maximum at 0



Starting at 0 only works when you **know** a value above 0 will show up. Integer.MIN_VALUE works for any int values.

WARMEST FREEZER READING

```
int[] temps = {-18, -21, -16};
int warmest = 0;
for (int t : temps) {
    if (t > warmest) {
        warmest = t;
    }
}
System.out.println(warmest);
```

PRINTS

0 (not even in the array)

CHANGE THE START TO

```
int warmest = Integer.MIN_VALUE;
```

Prints -16

A large, bold, teal-colored number '2' is centered within a white rounded square that has a slight 3D effect and a shadow.

PART 2

Autoboxing and unboxing

Autoboxing: into the envelope

AUTOBOX.JAVA

```
public class Autobox {  
    public static void showLaps(Integer n) {  
        System.out.println("Laps: " + n);  
    }  
  
    public static void main(String[] args) {  
        Integer laps = 8;  
        Double miles = 3.5;  
        System.out.println(laps + " " + miles);  
        showLaps(12);  
    }  
}
```

OUTPUT

```
8 3.5  
Laps: 12
```



Autoboxing turns an int into an Integer, or a double into a Double, when the value is

- 1. assigned** to a wrapper variable, or
- 2. passed** to a wrapper parameter.

Unboxing: out of the envelope

UNBOX.JAVA (MAIN)

```
Integer stored = -9;  
int copy = stored;  
int size = Math.abs(stored);  
Double rate = 1.5;  
double pay = rate * 10;  
System.out.println(copy + " " + size);  
System.out.println(pay);
```

OUTPUT

```
-9 9  
15.0
```

Where	Unboxed because...
int copy	assigned to an int
Math.abs	passed to an int parameter
rate * 10	used in arithmetic

Unboxing turns an Integer into an int, or a Double into a double. It mirrors autoboxing.

Only the matching wrapper

DOES NOT COMPILE

```
Double price = 5;
```

incompatible types:
int cannot be converted to Double

COMPILES

```
Double price = 5.0;
```

prints 5.0

COMPILES: UNBOX, THEN WIDEN

```
Integer n = 4;  
double x = n;
```

prints 4.0

DOES NOT COMPILE: WOULD NARROW

```
Double y = 2.5;  
int k = y;
```

incompatible types:
Double cannot be converted to int

Unboxing null



Unboxing needs an object to open. Unboxing **null** compiles, then crashes with a **NullPointerException**.

LOOKUP

```
Integer record = null;  
System.out.println("Checking record");  
int next = record + 1;  
System.out.println(next);
```

OUTPUT, THEN THE CRASH

```
Checking record  
Exception in thread "main"  
java.lang.NullPointerException:  
Cannot invoke  
"java.lang.Integer.intValue()"   
because "<local1>" is null
```

The first println runs. The crash comes on the next line, before next ever gets a value.

Box, unbox, or neither?

PAUSE AND TRY

For lines 1-4: autobox, unbox, or neither? Then: what prints?

```
1 Integer tickets = 40;
2 int left = tickets - 3;
3 Double cost = 2.25;
4 double total = cost * left;
5 System.out.println(total);
```

Line	Box, unbox, neither?
1	
2	
3	
4	

3

PART 3

Sealed for good

The envelope never changes

ALIAS

```
Integer points = 7;  
Integer saved = points;  
points = points + 2;  
System.out.println(points);  
System.out.println(saved);
```

OUTPUT

9
7

Wrapper objects are **immutable**.

AFTER `SAVED = POINTS;`

points



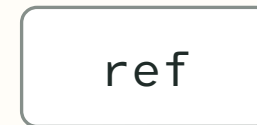
saved



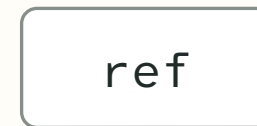
Integer object
value: 7

AFTER `POINTS = POINTS + 2;`

points



saved



new Integer object
value: 9

old object
value: 7

A method cannot change your Integer

TIP.JAVA

```
public class Tip {  
    public static void addTip(Integer bill) {  
        bill = bill + 5;  
        System.out.println("Inside: " + bill);  
    }  
  
    public static void main(String[] args) {  
        Integer total = 20;  
        addTip(total);  
        System.out.println("In main: " + total);  
    }  
}
```

OUTPUT

```
Inside: 25  
In main: 20
```



Assigning to the parameter changes only the parameter, and an Integer has no method that changes it. **main never sees 25.**

A large, white, rounded square containing a bold, teal number '4'.

4

PART 4

Text into numbers

Integer.parseInt and Double.parseDouble

JAVA QUICK REFERENCE

```
static int parseInt(String s)
static double parseDouble(String s)
```

OUTPUT

```
Lee: 43
12.0
```

ONE RECORD FROM A FILE

```
String line = "Lee,42,3.5";
String[] parts = line.split(",");
int points = Integer.parseInt(parts[1]);
double hours = Double.parseDouble(parts[2]);
System.out.println(parts[0] + ": " + (points + 1));
System.out.println(points / hours);
```

AFTER PARSING

Name	Type	Value
parts[1]	String	"42"
points	int	42
hours	double	3.5

parseInt does not round



Text that is not exactly a number of the right kind throws a **NumberFormatException**.

MILEAGE

```
String miles = "2.75";
System.out.println("Reading miles");
int m = Integer.parseInt(miles);
System.out.println(m);
```

OUTPUT, THEN THE CRASH

```
Reading miles
Exception in thread "main"
java.lang.NumberFormatException:
For input string: "2.75"
```

Text	Integer.parseInt	Double.parseDouble
"75"	75	75.0
"2.75"	crashes	2.75
" 14"	crashes	crashes

Number or text?

PAUSE AND TRY

Write the exact output, one line per println.

```
System.out.println(Integer.parseInt("12") + "3");  
System.out.println(Integer.parseInt("12") + 3);  
System.out.println(Double.parseDouble("4") / 8);
```

For each line, decide first: after parsing, is + joining text or adding numbers?

Words to know

Wrapper class

A class whose objects each hold one primitive value: Integer, Double.

Autoboxing

The compiler turning an int into an Integer, or a double into a Double.

Unboxing

The compiler taking the int out of an Integer, or the double out of a Double.

Immutable

Can never change after it is created. Every Integer and Double is.

parseInt / parseDouble

Static methods that turn a String like "42" into a number.

NumberFormatException

Thrown when the text is not a number of the right kind.

Box it, open it, parse it



Explain to your partner what happens, step by step, when Java runs `total = total * 2;` on an Integer total.

SENTENCE STARTER

First Java ____ total to get an int, then it doubles it, then it ____ the result into a ____ object.

Next: the Numbers in Envelopes puzzle worksheet.