

Boxing and unboxing

Stack

a →

int	1234
object	0

b →

Heap

Stack

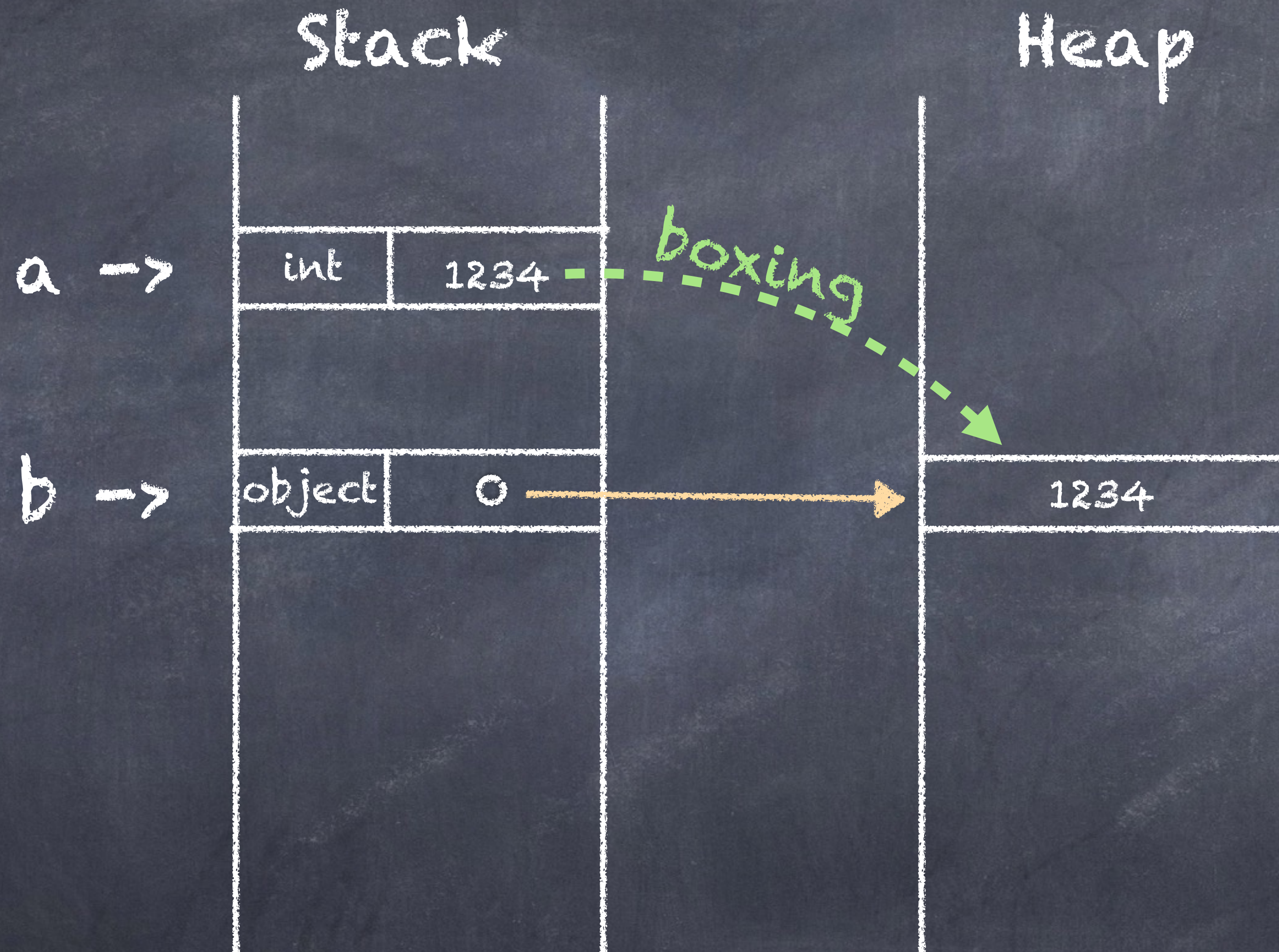
a →

int	1234
object	0

b →



Heap



Stack

a →

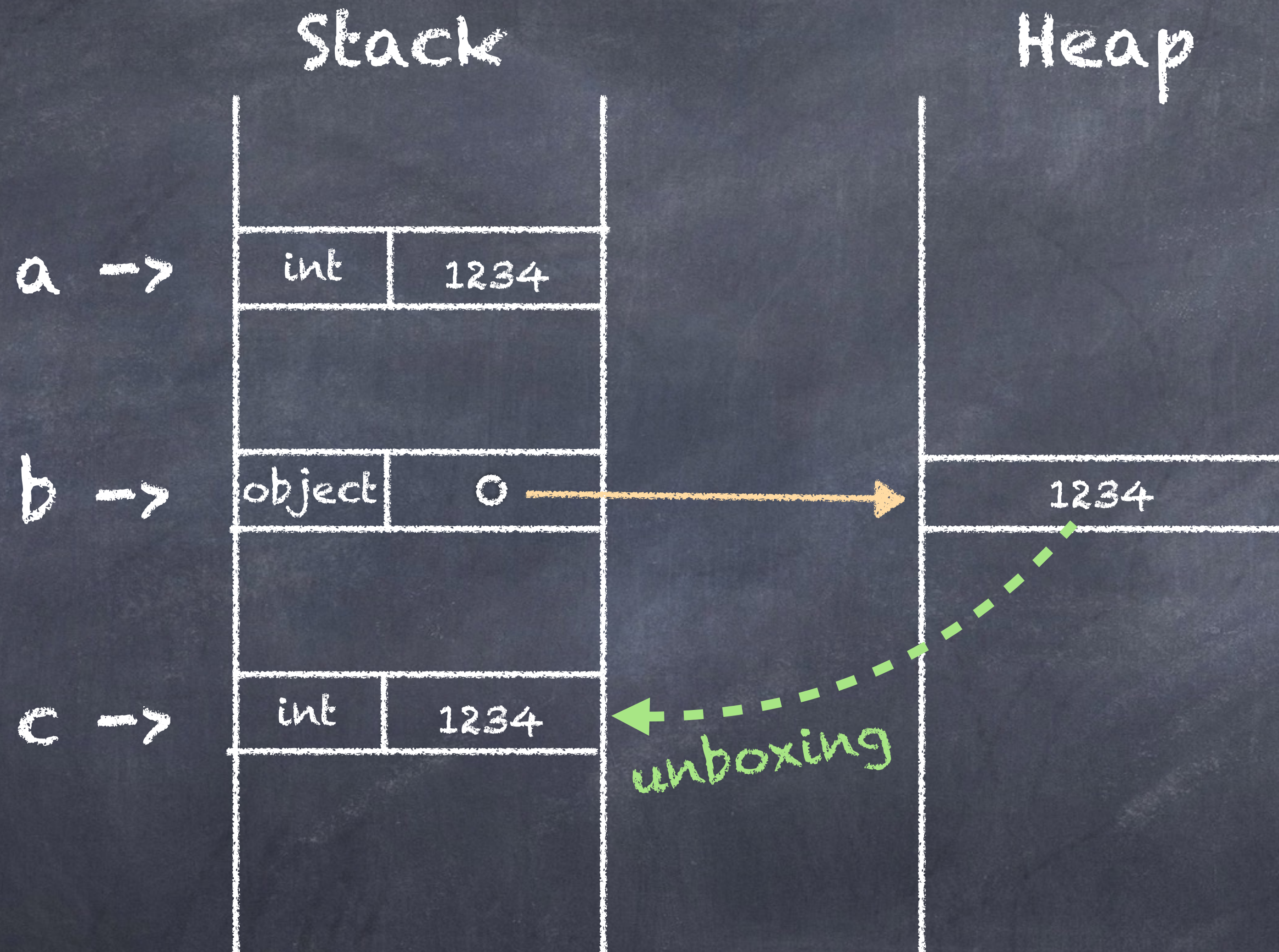
int	1234
object	0
int	0

b →

c →

Heap

1234



Boxing and unboxing

- Boxing takes a value type on the stack and stores it as an object on the heap
- Boxing happens when you assign a value type to a variable, parameter, field or property of type **object**
- Unboxing unpacks a boxed object on the heap, and copies the value type inside back to the stack
- Unboxing happens when you cast an object value to a value type
- Boxing and unboxing negatively affect performance