



# Compilers

---

## Java Coercions

- Java allows primitive types to be coerced in certain contexts.

*int float*

- In  $1 + 2.0$ , the *int* 1 is widened to a *float* 1.0

*→ int2float(1) + 2.0*

- A coercion is really just a primitive function the compiler inserts for you
  - Most languages have extensive coercions between base numeric types

- Java distinguishes two kinds of coercions & casts:
  - Widening always succeed (int → float)
  - Narrowing may fail if data can't be converted to desired type (float → int, downcasts)  $x:A$

$2.0 \rightarrow 2$   
 $2.5 \rightarrow$

$(B) \ x$

$A$   
 $1$   
 $B$

- • Narrowing casts must be explicit
- • Widening casts/coercions can be implicit

What is the only type in Java for which there are no coercions/casts defined?

Bool

- Coercions can lead to surprising behavior
  - Consider an example from PL/I
  - Let A, B, C be strings of 3 characters

B = '123' → 123  
C = '456' → 456  
A = B ⊕ C → 579 → '\_\_\_579' →  
'\_\_\_'

- What is A?