CSE 114A — lecture 10!

- Typeclasses

(+) :: Int → Int → Int ? no...

3 + 4 ✓

3.2 + 4.9 ✓

True + False x

(+) :: a → a → a ? no...

(==) :: String → String → Bool ? no...

"pikachu" == "venusaur" ✓

True == True ✓

3 == 70 ✓

(\x → x) == (\y → y) x

(==) :: a → a → Bool ? no...

Typeclasses address this issue.

"How to make ad hoc polymorphism less ad hoc"

↳ Wadler + Blott, 1989

Typeclasses — Haskell (since the 90s)

Traits — Rust

Interfaces — Java

Concepts — C++

Think of a typeclass as a set of operations that you can do on values of any type that implements that typeclass.

examples:

- for types that implement Num, you can do +, −, *, abs, etc.
- for types that implement Eq, you can do ==, /=.
- for types that implement Show, you can do show.