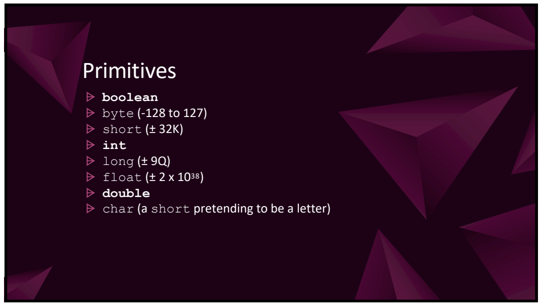
Java Primitives

CS 1400 - 2023
Cody Henrichsen

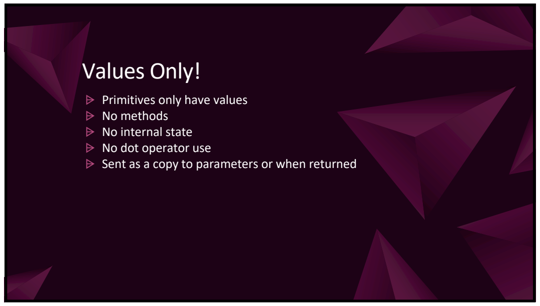
1



Primitives

- **`boolean`**
- `byte` (-128 to 127)
- `short` (± 32K)
- **`int`**
- `long` (± 9Q)
- `float` (± 2 x 10^{38})
- **`double`**
- `char` (a `short` pretending to be a letter)

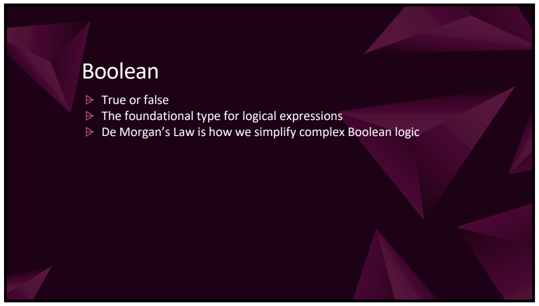
2



Values Only!

- Primitives only have values
- No methods
- No internal state
- No dot operator use
- Sent as a copy to parameters or when returned

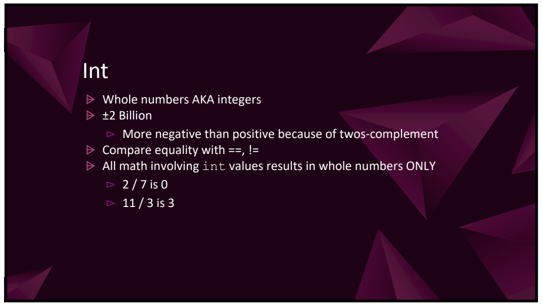
3



Boolean

- True or false
- The foundational type for logical expressions
- De Morgan’s Law is how we simplify complex Boolean logic

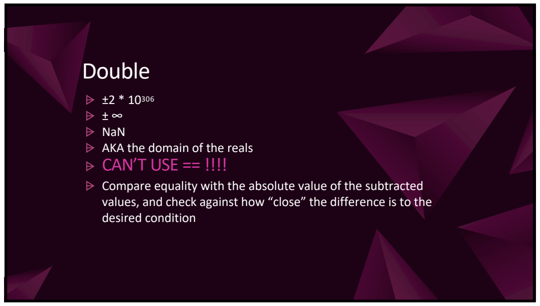
4



Int

- Whole numbers AKA integers
- ±2 Billion
  - More negative than positive because of twos-complement
- Compare equality with ==, !=
- All math involving `int` values results in whole numbers ONLY
  - 2 / 7 is 0
  - 11 / 3 is 3

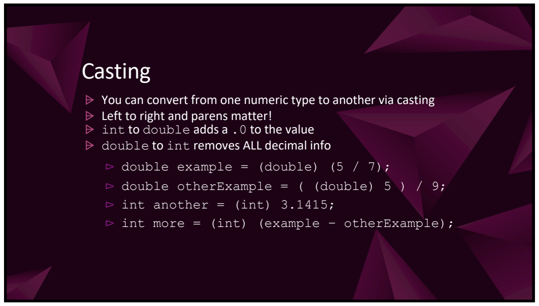
5



Double

- $\pm 2 * 10^{306}$
- ± ∞
- NaN
- AKA the domain of the reals
- CAN’T USE == !!!!
- Compare equality with the absolute value of the subtracted values, and check against how “close” the difference is to the desired condition

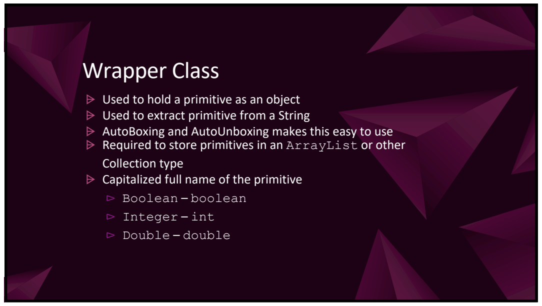
6



Casting

- You can convert from one numeric type to another via casting
- Left to right and parens matter!
- `int` to `double` adds a `.0` to the value
- `double` to `int` removes ALL decimal info
  - `double example = (double) (5 / 7);`
  - `double otherExample = ( (double) 5 ) / 9;`
  - `int another = (int) 3.1415;`
  - `int more = (int) (example - otherExample);`

7



Wrapper Class

- Used to hold a primitive as an object
- Used to extract primitive from a String
- AutoBoxing and AutoUnboxing makes this easy to use
- Required to store primitives in an `ArrayList` or other Collection type
- Capitalized full name of the primitive
  - `Boolean` – `boolean`
  - `Integer` – `int`
  - `Double` – `double`

8