#Core Java - chapter 3 Sections 3.3.1 + 3.3.2

11/30/22

3.3.1 Continued

advice: use int

Use int in most cases.
Short & byte are for specialized applications or where space is at a premium,
e.g. large arrays
Use long if you need the extended range

integer literals

Long suffix = L or l
Hex prefix 0x or 0X
octal prefix 0 → few people use these b/c confusing
binary 0b or 0B

underscores in literals

underscores in integer literals are ignored, e.g. 1_000
just for readability

no unsigned types

Java does∅ have unsigned versions of integer types.

working with signed as unsigned

unsigned : can treat integral types as unsigned
addition
subtraction
multiplication
~~division~~
} these will work unless overflow or underflow occurs

For other operations, call Byte.toUnsignedInt(b), then process it, then cast it back to byte.

→ Integer and Long classes have methods for unsigned division + remainder

3.3.2 Floating point types

floats range/storage

Type	Storage Requirement	Range
float	4 bytes	Approx ± 3.40282347 $\times 10^{38}$ (6-7 significant decimal digits
double	8 bytes	Approx ± 1.79769313486231570 $\times 10^{308}$ (15 significant decimal digits

Advice: which should I use?

Use float only if you are using a library that requires them or you need to store a lot of them. Otherwise, double.
literals : suffix f or F for floats. Otherwise doubles are assumed, but you can explicitly use d or D.

- carry over exercises: Write a program that returns 2
Put multiple classes in the same file. Is it allowed if at least 1 class matches filename? What if the other classes are private?