

COP 3330 1/14/26

- 1) Quick Review Course Website
- 2) HMK: 1 Download Java Compiler (JDK 25) + an IDE if you want, then do PI due on FRI (Fin Aid Asgn)
- 3) Hello4.java
- 4) Lec 2 - Variable, Arith Expr, Input

public static void main (String [] args) ??? 

args[0] ↑

↓ Command line

> java Hello4 Arup Guhu arguments

OPTIONAL

args[i]

Variables, Arith Expr, Input

types $[-2^7, 2^7-1]$, $[-2^{15}, 2^{15}-1]$, $[-2^{31}, 2^{31}-1]$

primitive types: byte, short, int, long, $[-2^{63}, 2^{63}-1]$

boolean, char

true
false

float, double

32 bits
6 digits precision
12 digits precision
real #s
64 bits

To declare:

int age, year; // Uninitialized

int money = 12; // Init.

double aGPA = 4.0, bGPA = 3.0;

Variable's scope (where it's valid) is within the nearest set of $\{ \}$ in which it's defined.

```
{
    :
    int age;
    :
    {
    }
    {
    }
    {
    }
}
```

→ after this age doesn't exist

Arith Expr

+ , - , * } Same as in all langs

/ } a/b uses integer division if a, b are ints

floating pt division if EITHER is a floating pt type.

$$14/3 \rightarrow (4)$$

$$199/200 \rightarrow (0)$$

$$-7/3 \rightarrow (-2)$$

} int div
lops off
decimal part

MOD

$a \% b$ } work exactly like C
remainder when a is divided by b

$$23 \% 7 \rightarrow (2)$$

$$199 \% 200 \rightarrow (199)$$

$$23 = \underbrace{(23/7)}_{\text{quotient}} * 7 + \underbrace{(23 \% 7)}_{\text{remainder}}$$

for mod if we get a negative
we might do something like

$$a = (a+n) \% n;$$

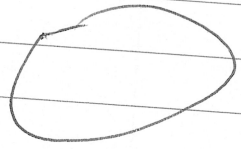
==

Reading Input

Scanner object — Java provides the Scanner class
(Old : Buffered Reader)

Scanner stdin = new Scanner(~~System.in~~);

class obj
 reference
 name
 Var

stdin → 

3 methods

next()
nextInt()
nextDouble()

method called on
stdin, returns
next integer from
input stream
↓

~~int~~ int age = stdin.~~next()~~ nextInt();

 ↑
Obj out
ref
name op