

OOPS. →

① ~~Classes~~ and ~~Objects~~.

② ~~Data Member~~

③ ~~Member function~~ → public

④ ~~Access Modifier~~ → private

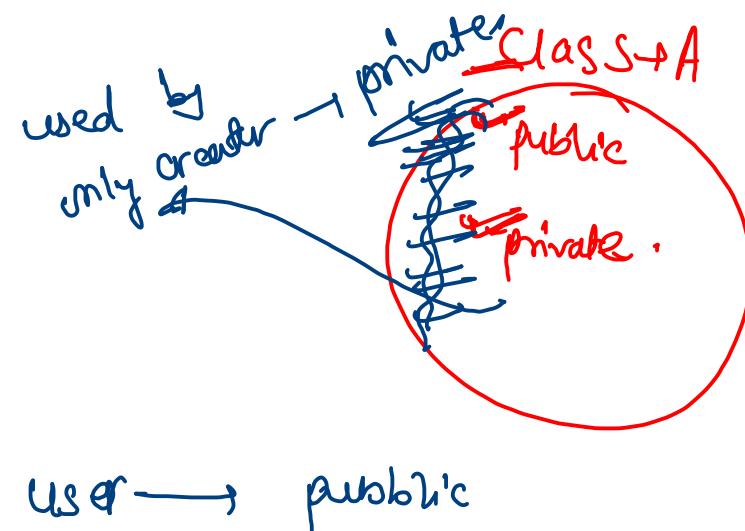
⑤ ~~Constructor~~ → Default
 ↳ (this) pointer parametrized

⑥ ~~Static~~

⑦ ~~String [] args.~~

⑧ OOPS game.

⑨ Creation of Stack & diff. function with same name but diff. arguments,
 → function overloading



Data types - C Primitive Data types)

- ① → `Byte` = }
- ② → `Short` = }
- ③ → `Int` = }
- ④ → `Long` = }
- ⑤ → `Float` = 6 }
- ⑥ → `Double` = }
- ⑦ → `Char` → \s }
- ⑧ → `Boolean` → `False` }
- .

Creation on
Stack is possible

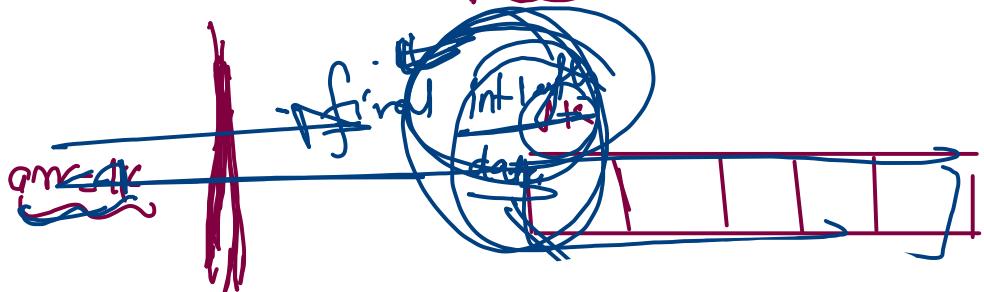
Rest all are non primitive.



Memory allocation → Heap

Return → Stack

`int[] arr = new int[s]`



User defined data type

Human

properties of Humans

map → with Human

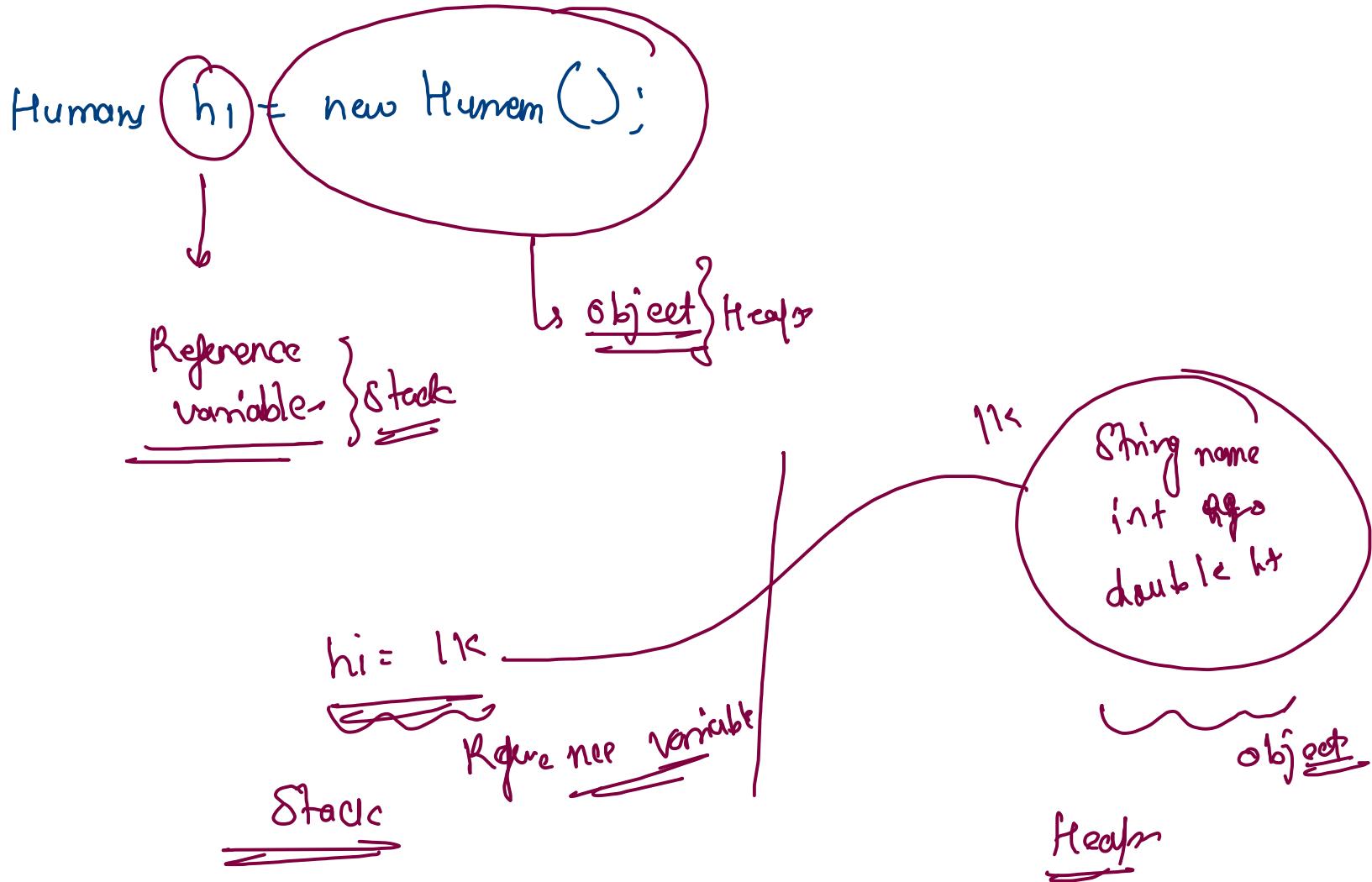
→ Name → String
→ age → int
→ ht → double

data members

Cars

properties of Cars.

→ company → String
→ color → String
→ price → int
→ Rating → double

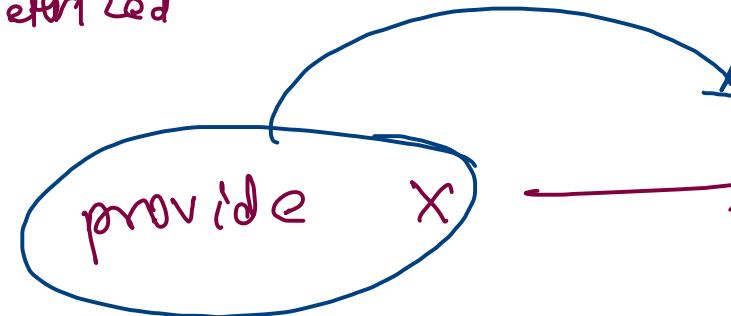


Constructor → Member function → without Return type
→ its name is same as class

default / empty constructor Name -

having
no arguments
constructor

parameterized



provide

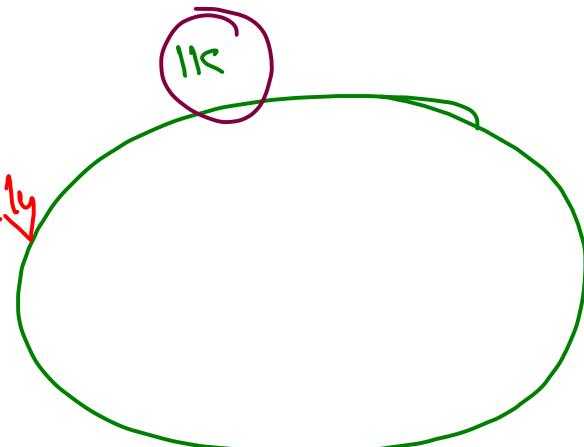
JVM remove its impact

this → self Referenced pointer



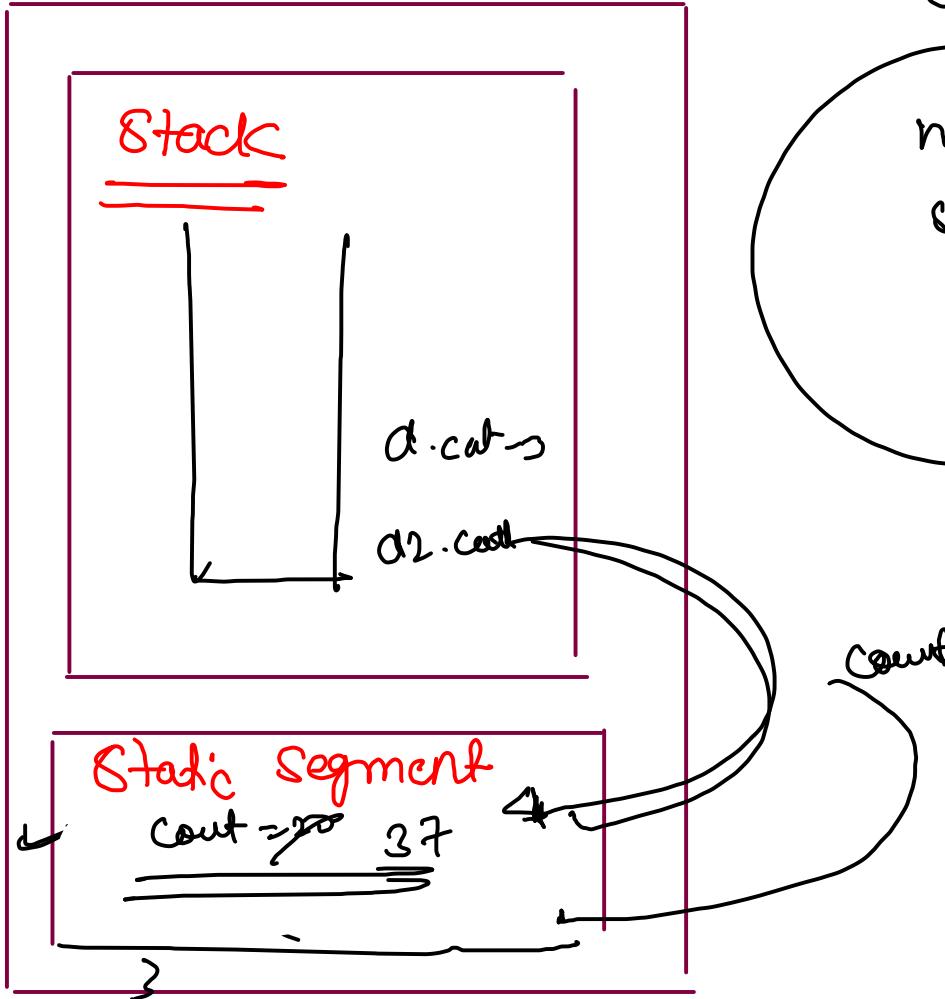
① this → self Referenced Pointer

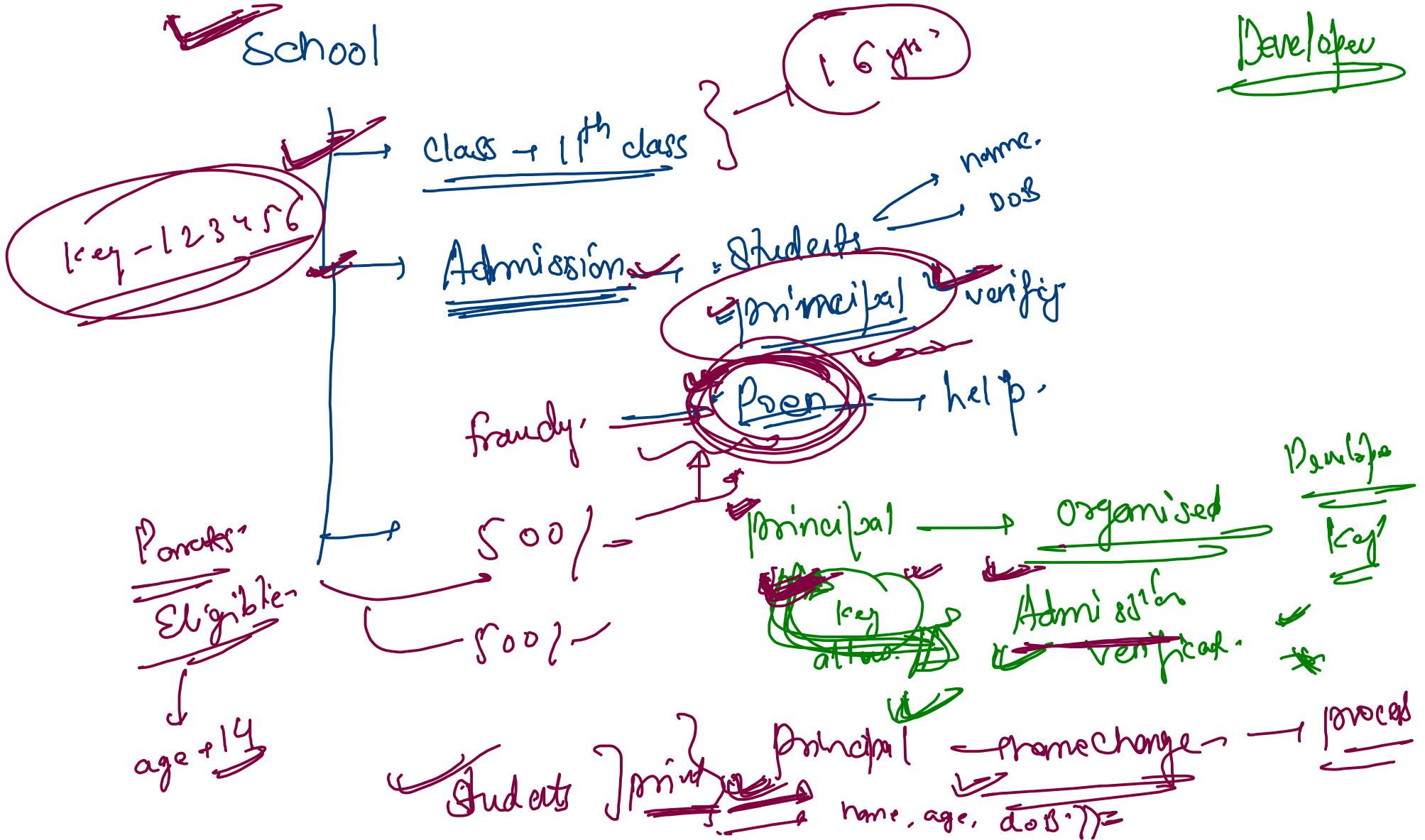
② this will pass automatically
with constructor as well as
with Member function



Static id count = 0
Static void func JS

3





OOPS

- ✓ Polymorphism
 - Run time →
 - override
 - overload.
 - Compile time →
- ✓ Encapsulation
- ✓ Abstraction
- ✓ Multithreading ← Synchronization
- ✓ Inheritance → Super
- ✓ Constructor → copy constructor
- ✓ this
- ✓ Interface →
 - ↳ Implementation
 - ↳ Usage
- ✓ Exception handling -

operator overloading

DP

↳ class

↳ Basic
DP

Saturday →

10 - 2

2:30 -

6:00

6:30 - 10:30

Sunday →

F