

- # OOPS
- ↳ Object Oriented Programming
- Classes
 - Object
 - Inheritance
 - Polymorphism
 - Encapsulation
 - Abstraction

Inheritance → Code reusability

class Animal {

legs;
eyes;

}

class Human {

{ class Human?
 { legs,
 eyes,
 ...
 ~
 → class Human extends Anime
 {
 print()
 }

→ child extends Parent

① When a child object is created, first Parent constructor is called, then the child constructor.

② this → refers the current object
 super → refers the parent object.

super = ...

super()

③ Java can only deal with default constructors on its own.

(7)

Constructors on its own.

→ Parent p = new child() ^{allowed}
Child c = new parent() ✗

Polymorphism

↓
one thing with
multiple behavior.

Polymorphism
├── compile time (Method overloading)
└── run time polymo. (Method overriding)

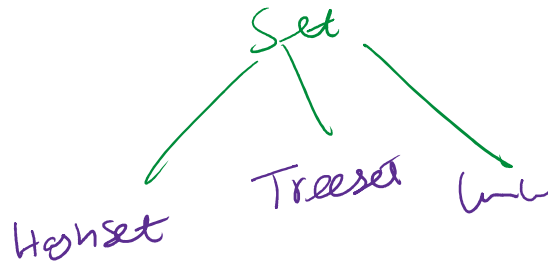
→ Return type doesn't help in method overloading.
only params decide that.

⇒ Method overriding
↓
Run time polymorphism

Run time 1 U

- Object decides the method
- Ref decides the variable

Set<Integer> s = new HashSet<>();
TreeSet<>()



Encapsulation

- Binding methods and variables together to have a better control on their access.

Access modifiers
| → private] →
...]

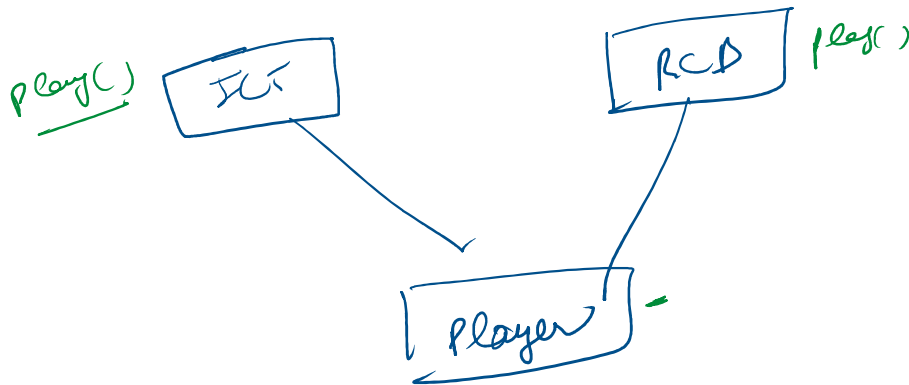
→ private
 → protected
 → default
 → public

→ private : only within the class
 → default : within the same package
 → protected : within the same package + child classes (can be in other package)

→ public : everywhere

* Abstraction → Interface → 100%
 → Abstract class → partial abstraction

⊕
 class extends class
 class implements interface
 interface extends interface



→ multiple Inheritance is not allowed in Java.

→ multiple interfaces can be implemented by a single class

⇒ Abstract class

abstract class X {
 {
 [abstract () ;]
 }
}

⊕ Abstract methods can only be inside an abstract class

static

static

final

final → primitive

→ Class: Cannot be inherited

→ method: Cannot be overridden

→ Variable: Value has to be initialized during the object creation and its value can not be changed

static

method: associated with class

variable: common to all the instance