Access specifiers inside java

→ they are known as access modifiers in java
→ they specifies access levels for class, methods or variables

there are 4 access specifier are there

Ⓐ Default
Ⓑ Public
Ⓒ Private
Ⓓ Protected

A class can have
Inner attributes
- (I) Variables
- (II) functions / methods
- (III) Class (inner class)

✱ we can use access modifiers on
✱ For inner attributes you can use any access modifier but for outer class you can use only default & public

```
class Demo {
int a=10;
Void meth() {
}
class inner {
}
}
```

→ for these you can use anything

→ only public & Default can be used.

How can you use your class?
⇒ you can create object
⇒ you can extend class to form inheritance

✱ has A & is A

✱ Inheritance lead to is A relation

✱ object creations leads to has A relation.

has A

```
class AA{
}
```

```
class zzz{
AA a=new AA();
}
```

```
class BB extends AA{
}
```

class zzz has A object of class AA

Class BB is A child of AA

See two packages

Package A	Package B	Err

class P_1 {
 —11—
 —11—
 }
	class Q_1 {
	 P_1 a = new P_1();
	 }
		for P_1 & P_2
		 P_2 is subclass of P_1
		 so P_2 ~~shar~~ ~~has~~ is subclass of P_1 in same Packet

class P_2 extends P_1 {
 —11—
 —11—
 }
	class Q_2 extends {
	 —11—
	 —11—
	 }	

from above we can infr some 4 relations

① Q_1 is using P_1 so Q_1 & P_1 share non-subclass Relation
② Q_2 ~~has~~ is A subclass Relation of P_1 in ~~dif~~ of is inter packet relation
③ Q_1 & Q_2 are not related at all. (same packet non subclass)

so we can draw table for that

~~So~~	Default	Private	Protected ~~Publ~~	Public
Same class	✓	✓	✓	✓
Same packet sub class	✓	X	✓	✓
Same packet non-sub class	✓	X	✓	✓
Different packet subclass	X	X	✓	✓
~~So~~ Different Packet non-sub class	X	X	X	✓

It easy table just try to brainstrome

⇒ Default will allow permission only for same packet

⊕ for default class from same packet can access anything but from different ~~class~~ packet it will not

⇒ you can only access ~~from~~ private members if you are in same class else no way possible

⇒ Protected is keyword which allow inheritance to happens so for ~~no~~ different packet subclass relation protected will able to access

⇒ Protected will decline for non subclass relation from different ~~p~~acket

⊕ Trick Imp

Default ⇒ only for packet itself
Public ⇒ available to all
Protected ⇒ self packet or inheritance (subclass ~~Re~~relation
Private ⇒ no entry except same class

Redraw table

	Default	Private	Protected	Public
Same class	✓	✓	✓	✓
Same packet subclass	✓	X	✓	✓
Same packet non-subclass	✓	X	✓	✓
Diff. Packet sub class	X	X	✓	✓
Diff packet non-subclass	X	X	X	✓

⊕ Follow above table and you will get all possible cases for Access specifier