2.11 - OOPS - 2

3:10 PM

Encapsulation

↳ Binding data members and methods together inside a single class to achieve desired level of access / security.

⇒ Access specifiers / modifiers

- → private
- → default
- → protected
- → public

1) private : accessible inside the same class.

⊛ We generally try to make our member variables private.

→ making getters and setters give us granular control over our variables.

→ default : accessible inside the same class + same package

package → collection of classes / interface.

⇒ protected : default + child class in a diff. package

⇒ public : anywhere in the project.

Polymorphism] → it is applicable on methods

$\underbrace{\text{Poly}}_{\text{many}}\underbrace{\text{morphism}}_{\text{forms}}$

- → Compile time polymorphism (Method over loading)
- → Run time " (Method Overriding)

$$\text{compile time} \leftarrow \underbrace{\underbrace{\text{Person p}}_{\text{ref variable}}}_{\text{Stack (static)}} = \underbrace{\underbrace{\text{new Person()}}_{\text{Object}}}_{\text{heap (dynamic)}}; \rightarrow \text{Run time}$$

⇒ Compile Time polymorphism
⇓
method overloading
↓
a class has multiple func^n with same name

a class has multiple func with same name
but diff. $\underbrace{\text{parameters}}_{\text{number/ data type.}}$

(*) Method overloading only works on parameters but not on return type.

⇒ Run time Polymorphism/ method Overriding

(*) Parent p = new Child() ✓ allowed
Child c = new Parent() ✗] NOT allowed.

(*) Its the object which decides which method has to be called.

> Methods are decided by object
variables are decided by reference.

Abstraction
↳ hiding the implementation and only showing the details.

partial abstraction

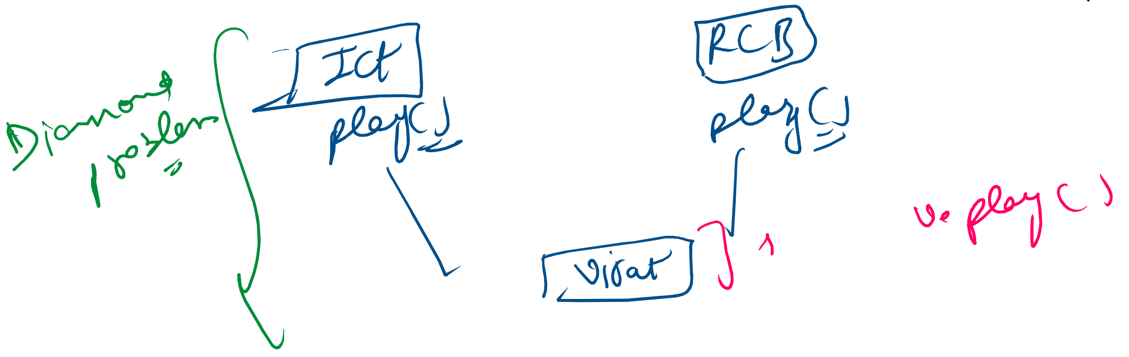
the ...

- → Abstract class } → partial abstraction
- → interface } → 100% abstraction

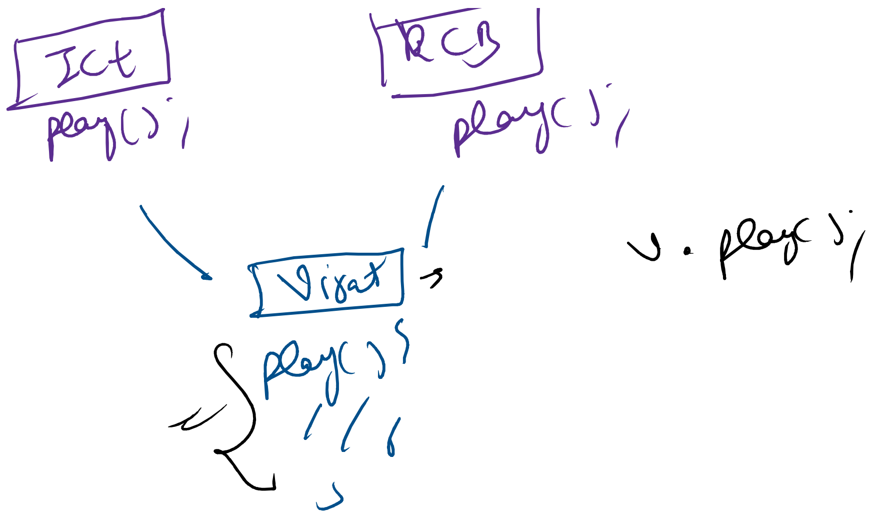
# class extends class
class implements interface
interface extends interface.

→ A child class has to implement all the methods of the interface.

# a class cannot extend more than one class. (No multiple inheritance)



Ⓐ Interfaces allow multiple inheritance



⇒ Interfaces cannot be instantiated (object).

abstract class
- → abstract (no impl)
- → non-abstract (imple)

→ abstract class cannot be instantiated.

Multi level Inheritance

$$\begin{tikzcd} A \arrow[d] \\ B \arrow[d] \\ C \end{tikzcd}$$

```
class A {
  int x;
  f1();
}
class B extends A {
  int y;
  f2()
}
class C extends B {
```

int x,
int y
f1()
f2()

```
class C extends B )
    int x;

}
```