5.2 - OOPS Revisit - 2

Saturday, October 25, 2025 10:31 AM

Inheritance → code reuse

Parent class ← class Person { name; age; }

Child class ← class Student extends Person { course; }

class Stu { name age course }

class Employee extends Student { }

When we make an object of child class, first the parent constructor is called.

⇒ super refers to the parent object.

Encapsulation

↳ Binding data members and methods operating on them in single class

operating on them in single class to achieve granular level of

Security

↳ Access modifiers

- private
- default
- protected
- public

⇒ private: can only be accessed inside the same class.

⇒ default: inside the same class and same package

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

- public: anywhere

Polymorphism

↳ forms

Polymorphism

many forms

- compile time (method overloading)
- run time (method overriding)

⇒ Method overloading
↳ same name, diff parameters.

⇒ Method overriding
↳

```
Parent p = new Child();
Child c = new Parent();
```

Abstraction

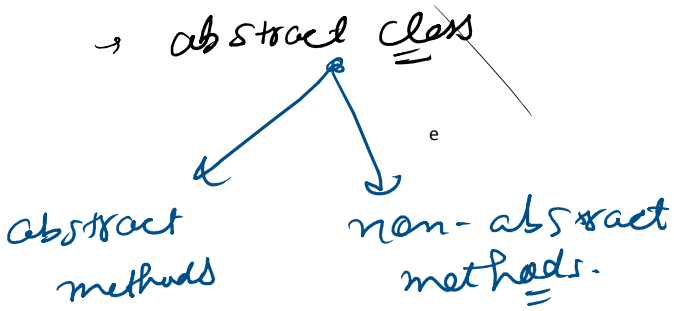
↳ hiding the implementation and showing only the necessary details.

- interfaces] → 100% abstraction

- interfaces → 100% abstraction
- abstract classes → partial abstraction

interface
↳ only the func^n sig nature
but not the implementation

(#) Interfaces are never instantiated.



(#) Abstract class cannot be instantiated.

(#) Even if there is a single abstract method, class has to be made abstract.

static keyword

- is the share the properties across objects
- static attributes / methods belong to the class not objects.
- static DOESN'T mean variable is immutable

final

- final variables are nothing but normal variables whose value has been fixed.
- final methods can not be overriden.

H.W

- IntelliJ community
- github account → website

→ github account → website

→ git → software

→ MySQL server

→ Workbench / DBeaver