

Agenda

①

Prototype

Registry

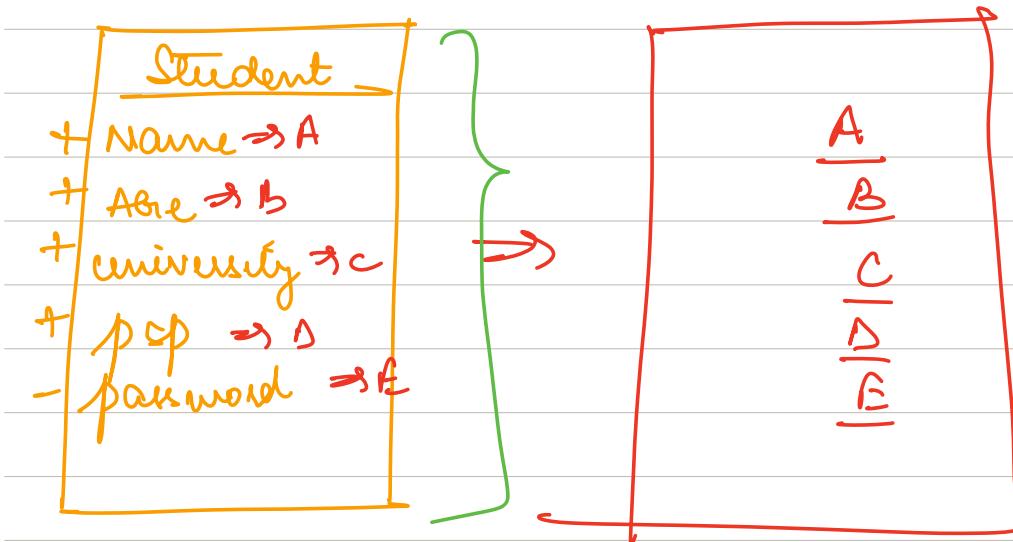
Design Pattern

PROTOTYPE DESIGN PATTERN

Problem Statement

- ① We already have an object of a class.
- ② We want to create copy of the object

→ Another object of the same class with exact same value of all the attributes as the original object



Salⁿ 1

Create a new object of the class and copy all the attributes from the original object

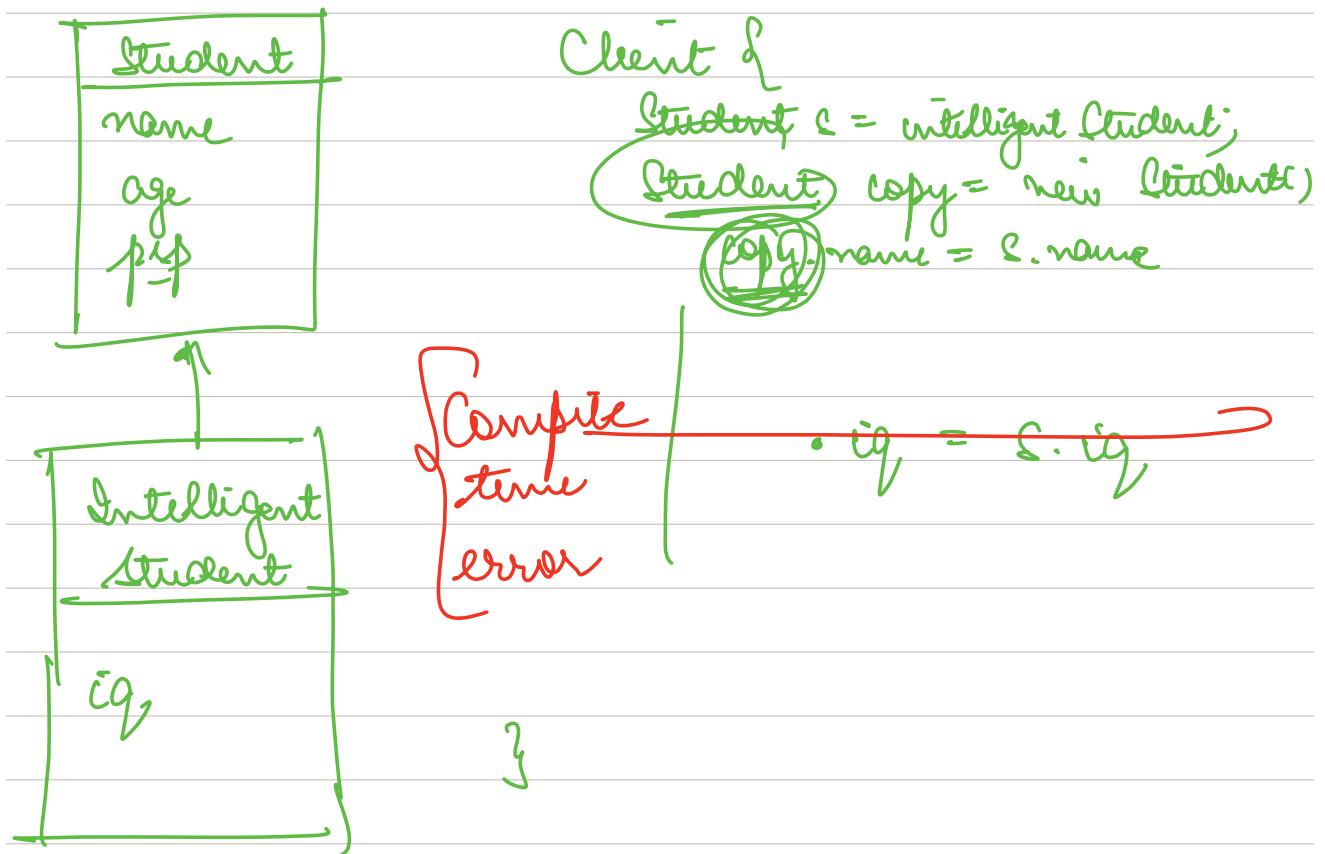
{ Student studCopy = new Student()
studCopy.name = Original Name;
studCopy.age = age;
studCopy.univ = univ;

Problem Statement

- ① Tight Coupling b/w Student and Client
→ Whenever any change happens in student class, I will have to make changes on the Client
- ② If there are private attributes, copy isn't possible

③

Client {
 psvm() {
 Student s = ; // Student or a child Student class.
 Student copy = new Student();
 copy.name = s.name;
 copy.age = s.age;
 }
}



Client {
`psvm() {`
`Student s =` ⇒ Violates O/C
 and SAP
Principle
`if (s instance of Student) {`
`IntelligentStudent iS = new dule`
`} copy (`)
`}`
`}`
`}`

SOLUTION 2 : COPY CONSTRUCTOR

Client {

psym() {

Student s = _____;

if (s is instance of IntelligentStudent){
Student copy = new IFS(s)}

⇒ Violates D/C
and LRP

↳ copy (s is instance of Student){
Student copy = new S(s)}

}

IDEAL SOLUTION

→ Object itself provides a method
to create a copy of the object

Client {

psym() {

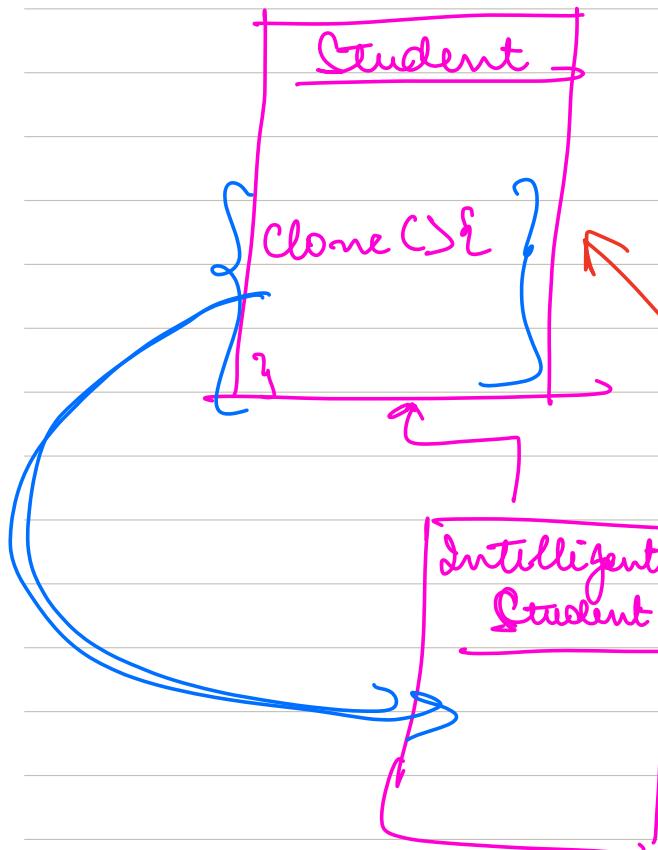
Student s = _____;

Student copy = s. copy()

s. clone()

}

→ clone()



→ If I forgot to implement clone in child, it will call parent's clone ⇒ THIS IS WRONG

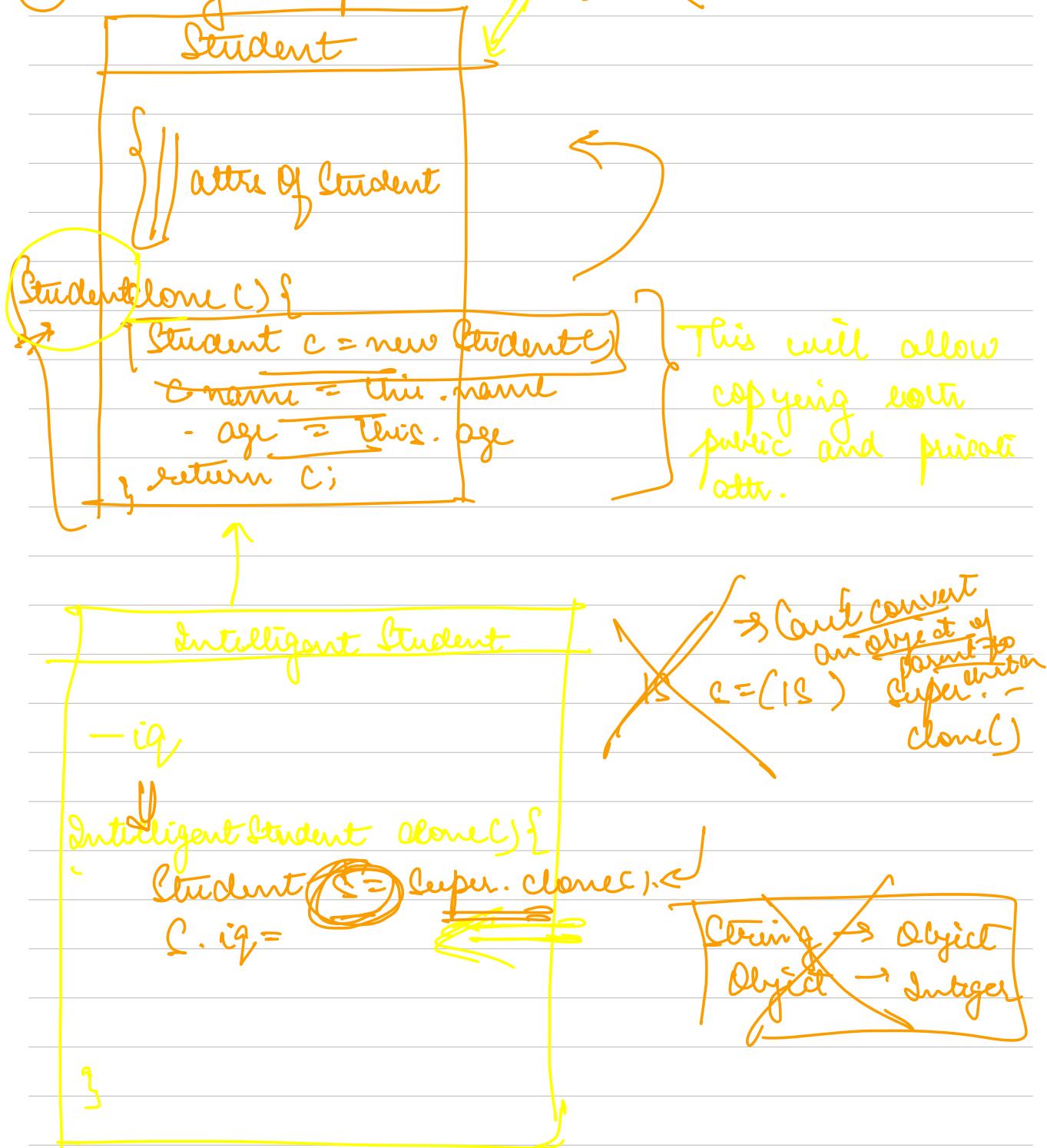
Client {
 Param() {
 Student s = new
 SC()
 }
}

It will only copy parent attributes
⇒ THIS IS WRONG

→ If any class implements the clone() method, all child classes of that class must also override the clone method.
Else it can lead to errors.

HOW TO IMPLEMENT clone() method

① Using super.clone() X



② Rewrite clone in child ~~X~~ will not work

Intelligent Student {

Intelligent Student clone () {

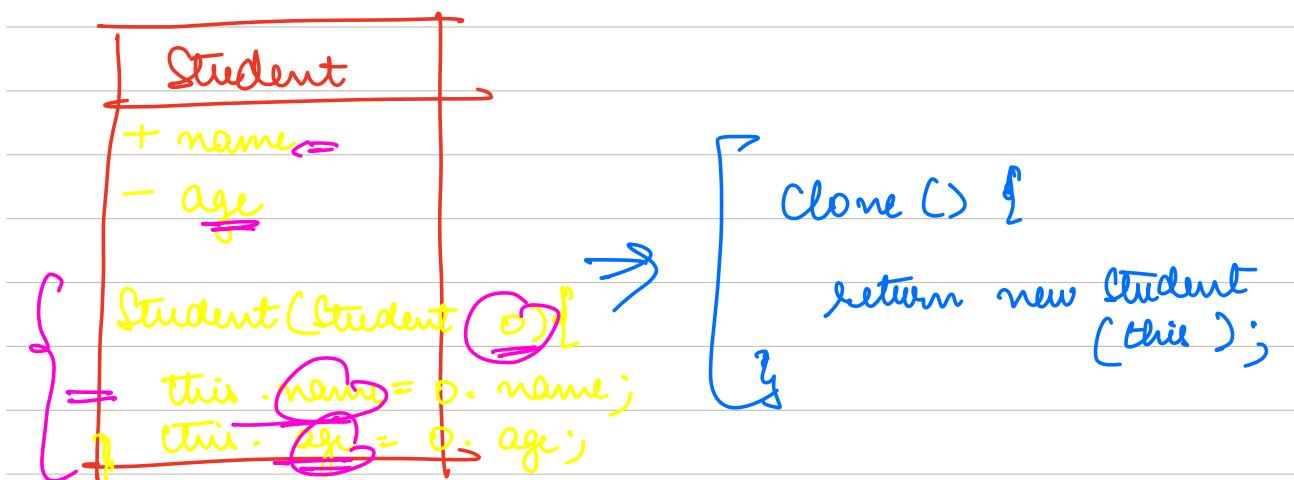
Intelligent Student is = new ICS;

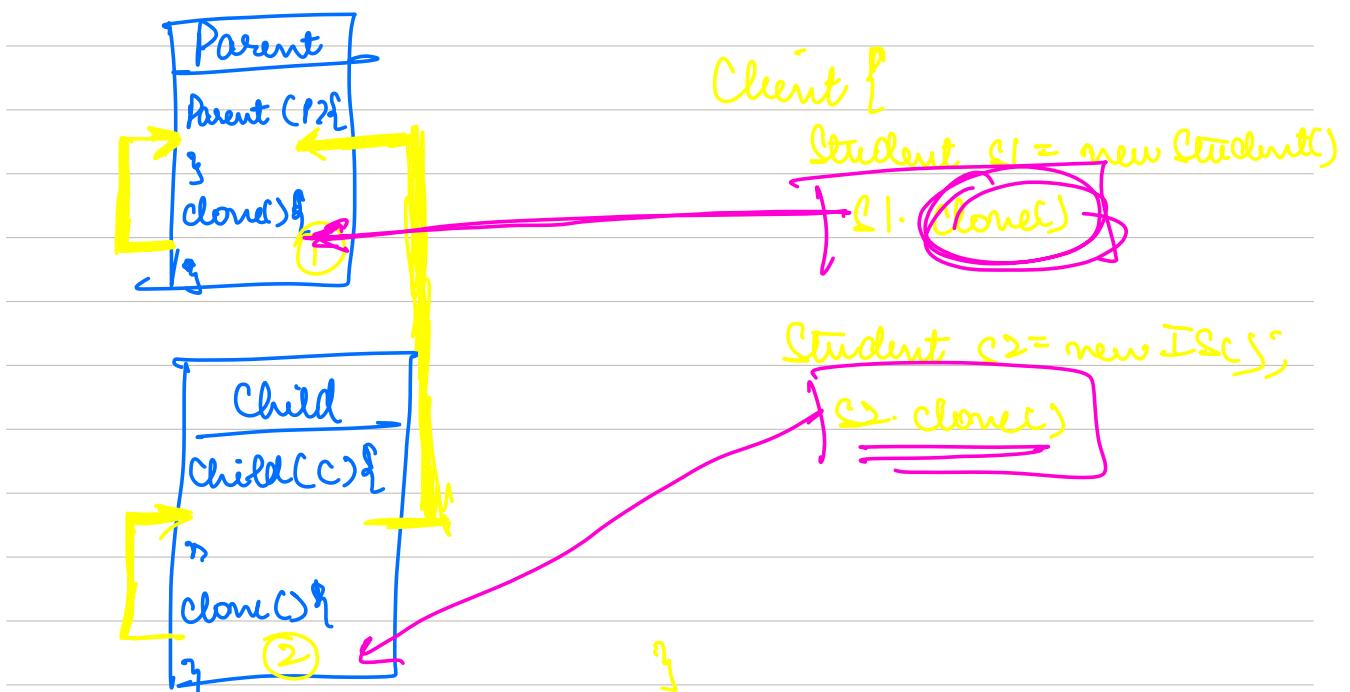
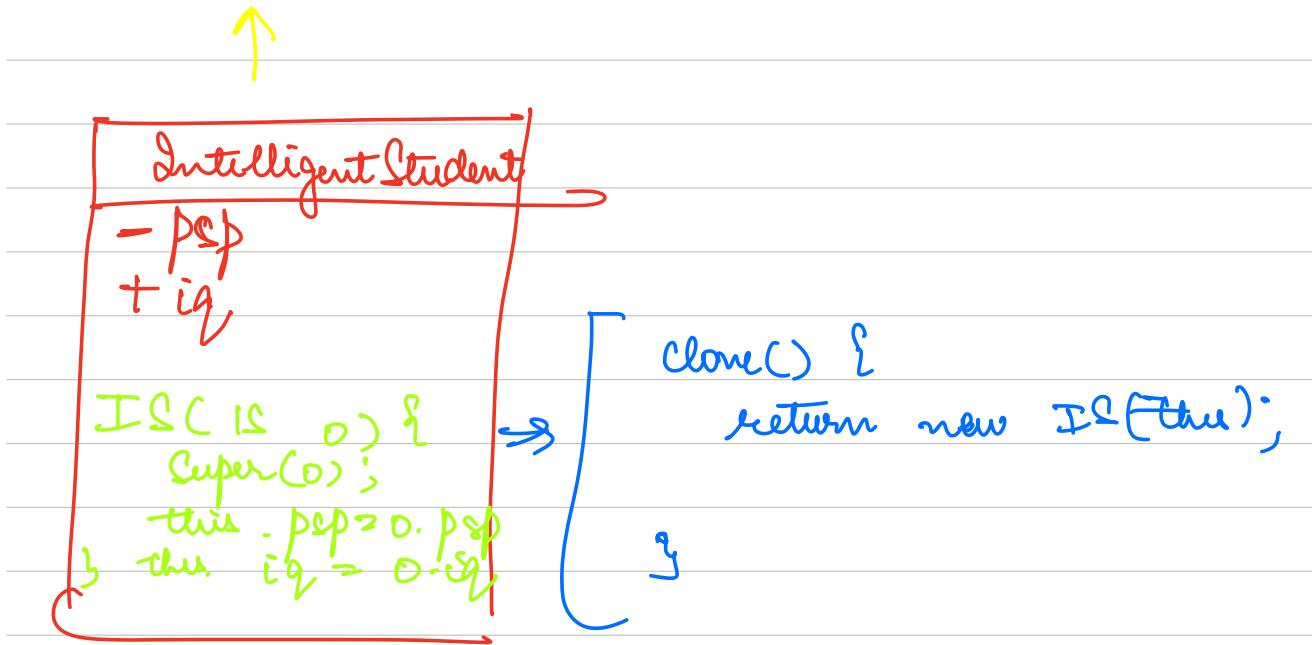
→ Copy all public
+ private vars
of self as
well as parent

→ But, I can't access private
atts

}

CORRECT SOLUTION : Clone () method
does nothing but
calls the copy
constructor





→ What is the best way to create a copy

→ But, why are we even creating a copy?

HTTP API

→ Building a weather app

weather.com

→ user enters loc^m

→ app shows weather at that loc^m

GET weather.com/api/weather/ location

{

Weather Details

}

We are using Google Search API

① get Search Results for A Query

GET google.com/search

{

type: SearchResults

query: { }

3

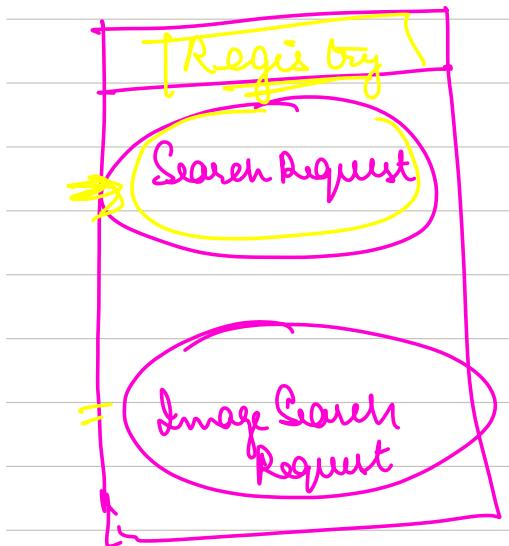
② get images for a query

```
GET google.com/search  
{  
  - type: IMAGE  
  - query:  
}
```

Client {

```
psym() {  
  Request r = new Request();  
  r.setUrl("google.com/search");  
  r.setParam("type", "Search Results");  
  r.setParam("query", " ");  
  r.send();  
}  
}
```

→ Across codebase I am fetching results from multiple places.



Client {

params) {

Request = X.get("SearchReq")

x.setQuery()

x.send()

②

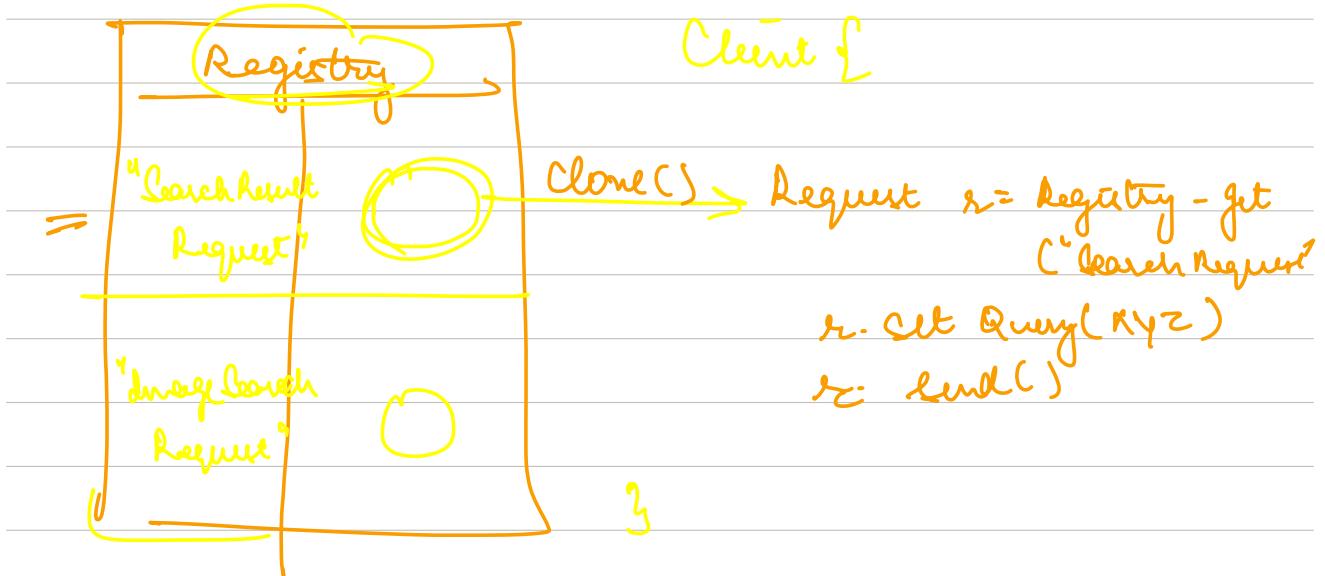
} }

⇒ We get a copy of an object
and we just set the dynamic
attr -

⇒ I have a registry class

⇒ Purpose is to store different
prototype

⇒ Whenever you need a particular
type of obj you call the registry.
⇒ Registry creates a copy of
the prototype and gives to you



Classmate Notebooks

- ① Cover Page
- ② Facts / Trivia Page

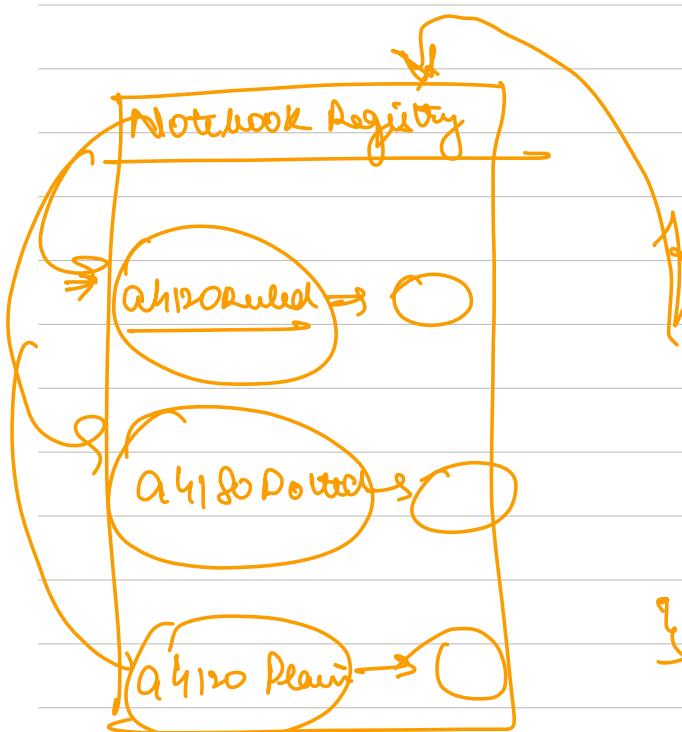
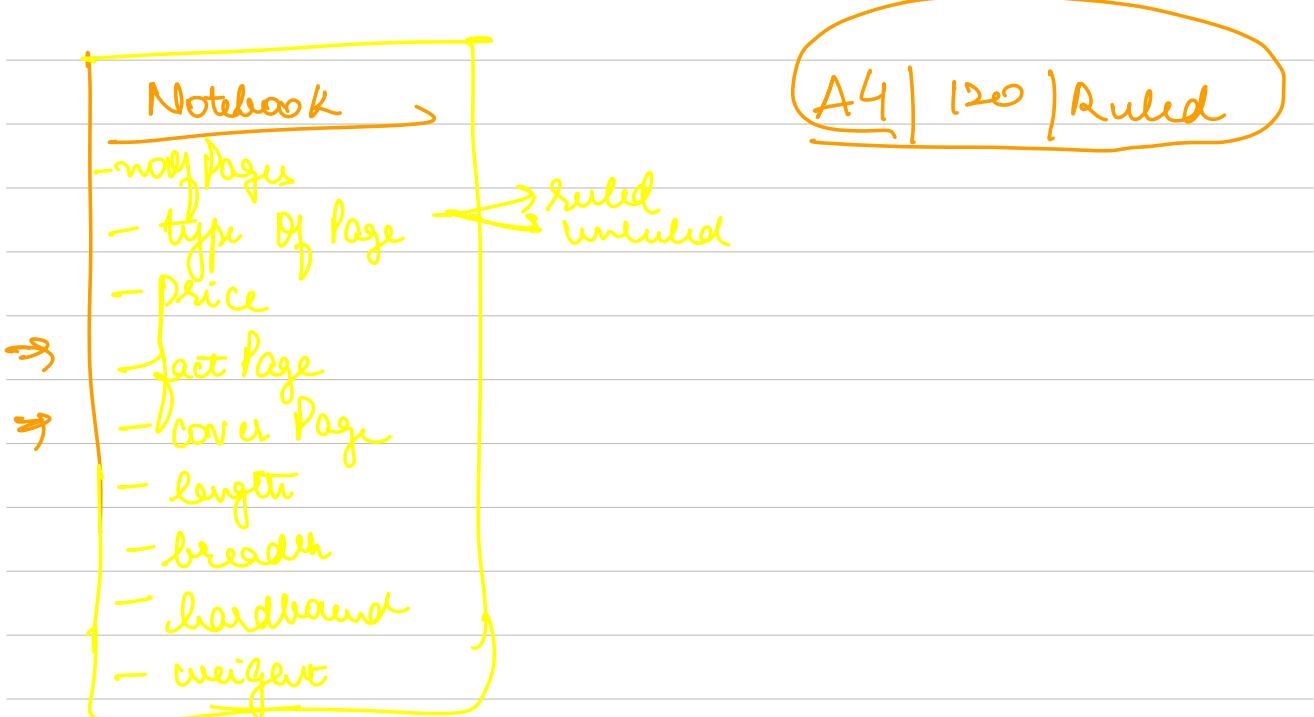
Types of notebooks that Classmate produces-

①	A4	120	Ruled
②	A4	180	Dotted
③	A4	180	Blank

Case Study \Rightarrow We are a SWE working at Classmate. Factory Management System

```
for (int i=0; i<1000; ++i) {
    produce Notebook;
```

)



```

for (int i = 0; i<1000; ++i)
    String type = get input()
    =Notebook nb = reg.get (type)
    nb.set Cover (random)
    nb.set Facts (random)
    nb.produce ()
}

```

```

    } Class NotebookRegistry {
        Map<String, Notebook> map;

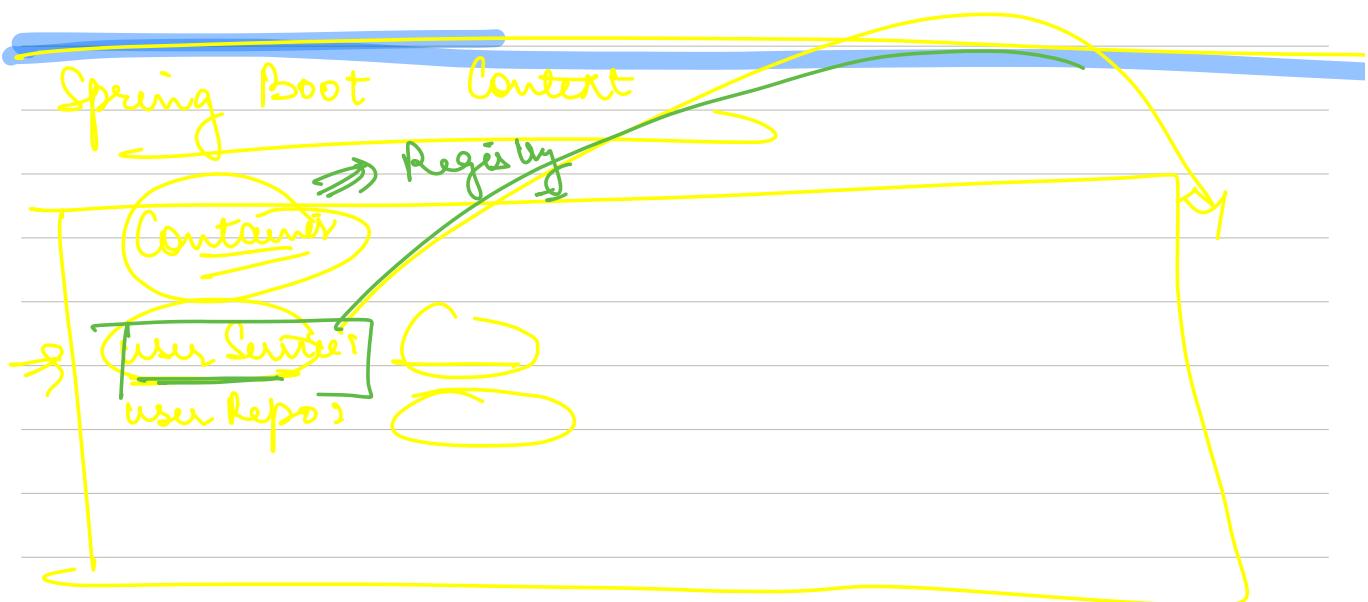
        void register (String name, Notebook nc) {
            map.put(name, nc)
        }

        Notebook get (String name) {
            return map.get(name).clone();
        }
    }

```

Prototypes \Rightarrow Object for which we can
create a copy

Registry \Rightarrow Storage for objects & will
need later



H/W

① Read and implement prototype
registry from refactoring.guru