

Agenda. Creational  
⇒ Prototype & Registry.

Problem Statement

- ⇒ Given an object of a class.  
⇒ We need to create a copy of this object  
(create a new object in memory with exact  
same attrs).

psum 1) C

st1

Student st2 = new Student();

st2.name = st1.name;

st2.age = st1.age

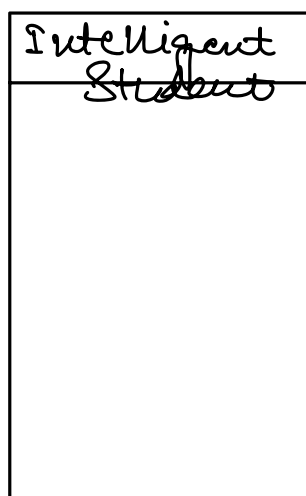
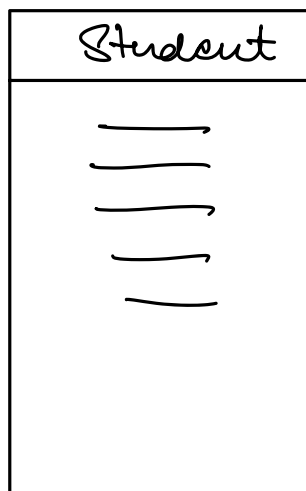

|||

Cons.

1) lot of code.

2) There can be some private fields which we  
might not be able to access.

#



OCP:

```
Student st = Student obj /  
IntelligentStudent  
Obj
```

```
Student copy;
```

```
{  
    if (st instanceof Student) {  
        Copy = new Student();  
    }  
    else if (st instanceof IS) {  
        Copy = new IS();  
    }  
}
```

Copy Constructor.

```
Student {
```

```
    Student (Student st) {
```

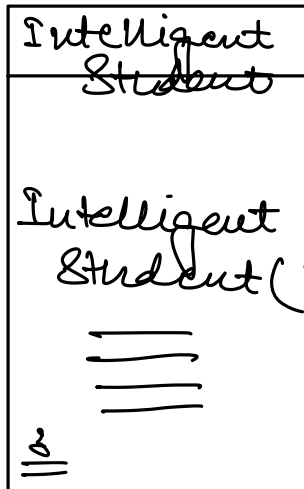
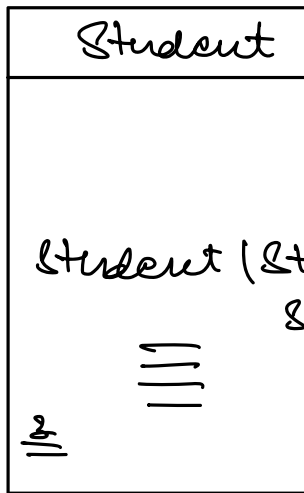
```
        this.name = st.name;
```

```
        this.age = st.age;
```

```
    }  
}
```

2

Student copy = new Student(st);



Student st = Student obj /  
IntelligentStudent  
Obj

Student copy;

if (st instanceof Student) {  
    copy = new Student(st);

3  
else if (st instanceof IS) {

    copy = new IS(st);

3

⇒ If a client wants to create a copy of an object, having the logic of creating a copy on client is prone to Errors.

⇒ Ideal sol<sup>n</sup> is to provide a simple copy method to client and all the complex logic should be handled by our code.

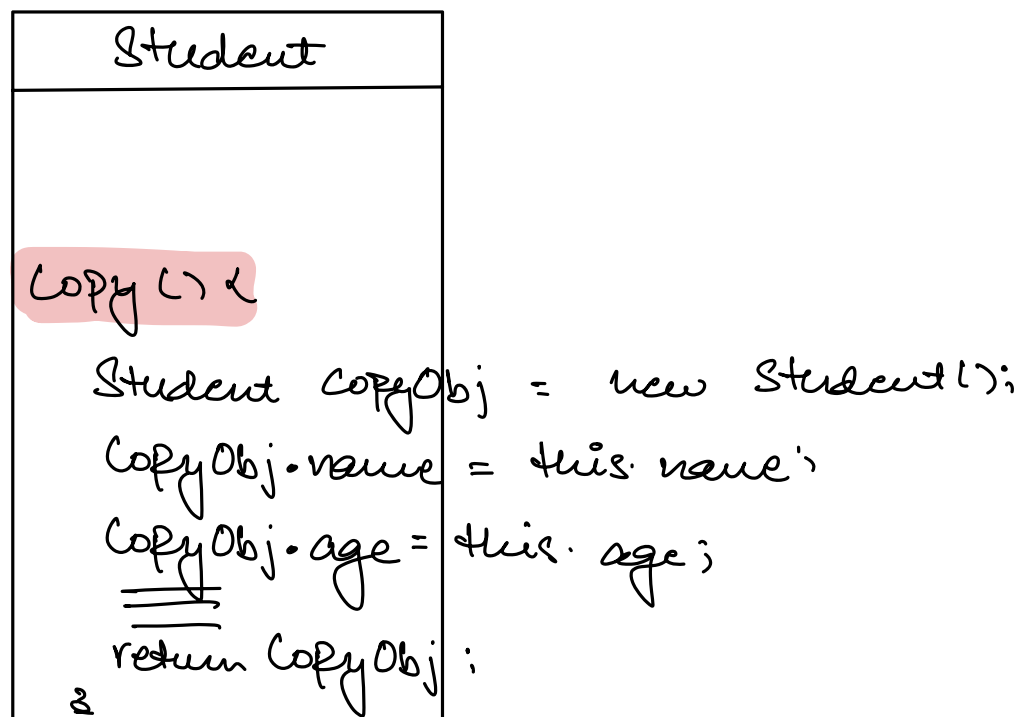
Student st = Student obj /  
IntelligentStudent  
Obj

Student copyObj = st. copy()

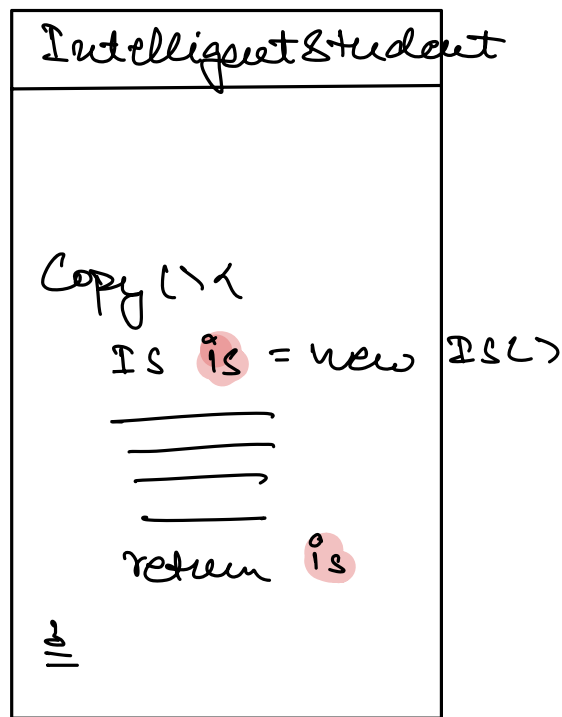
Progs

1) No DCP Violation.

2) Not all the internal details are exposed to  
Client.



⇒ Runtime  
Polymorphism



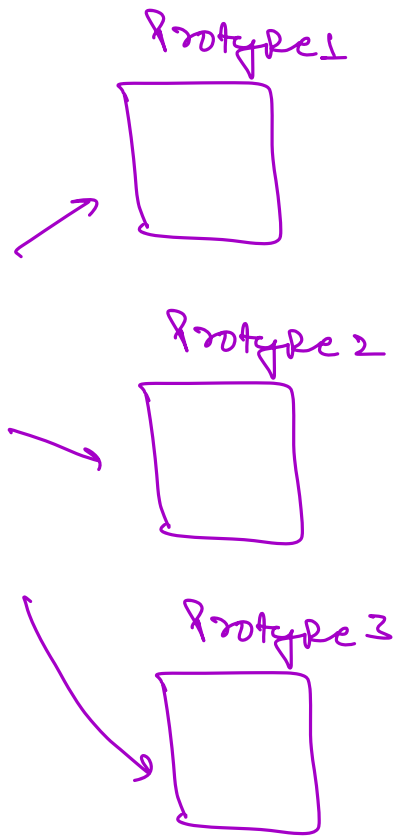
Note: All the child classes must override the  
copy method, otherwise it can lead to  
unexpected results.

PROTOTYPE.

↳ Blueprint / copy

⇒ Classmate Notebooks.

<u>Notebook.</u>	
width	✓
height	✓
no of pages	✓
type	✓
mrp	✓
frontPage	✗
<u>lastPage.</u>	✗



Requirement: Create 100,000 notebooks of type ruled with 100 pages of size A4 size.

Prototype Object.

<u>A4-ruled-Nb</u>	
width	= 100
height	= 50
no of pages	= 100
type	= Ruled
mrp	= <u>60 Rs.</u>

← 100,000 copy objects.

# Prototype Design Pattern

Often there are scenarios where we need to create lot of objects with only few different attrs and lot of same attrs, so we can create a Prototype object with common Properties & other objects can be created as a copy of this Prototype object & just set the object specific attrs.

⇒ Search API's

<https://www.google.com/search?query=India>

Way 1

```
SearchAPI api = new SearchAPI();  
api.setProtocol("https")  
api.setDomain("google.com")  
api.setType("search")  
api.setQuery("India");
```

```
SearchAPI api2 = new SearchAPI();  
api2.setProtocol("https")  
api2.setDomain("google.com")  
api2.setType("search")  
api2.setQuery("World Cup");
```

Way 2

Registry

```
SearchAPI prototype =  
    new SearchAPI();  
prototype.setProtocol("https")  
prototype.setDomain("google.com")  
prototype.setType("search")
```

```
SearchAPI api = prototype.copy();  
api.setQuery(——);
```

Prototype

⇒ When we have scenario of creating of lot of object with only few different properties.



Student i

- name
- age
- univName
- batchName
- avgBatchSp

3

way-1 X

Student Sumit = new Student()

Sumit.name = "Sumit"

Sumit.age = 25

Sumit.univ = "Scaler".

=====

way-2

⇒ Create 1 sample | prototype obj for each batch.

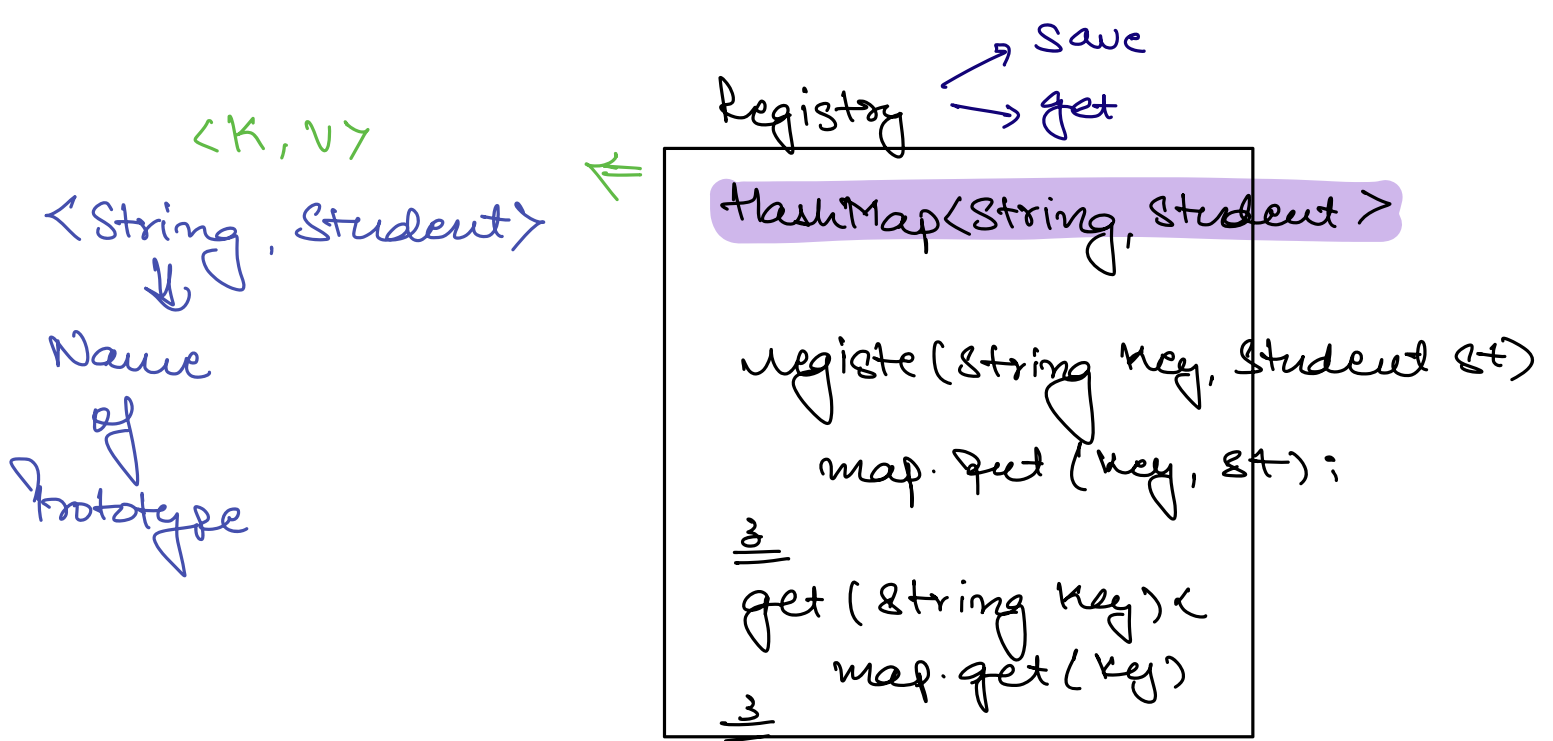
Student Mar24MWFLDStudent =  
new Student();

Mar24MWFLDStudent.batch = "Mar24MWFLD"

Mar24MWFLDStudent.avgBatchSp = 70

Mar24MWFLDStudent.univName = "Scaler"

⇒ Store this object for future use.



```
Student Mar24TTSLDStudent =
    new Student();
```

```
Mar24MWFLDStudent.batch = "Mar24TTS LLD"
```

```
Mar24MWFLDStudent.avgBatchSp = 65;
```

```
Mar24MWFLDStudent.univName = "Scaler"
```

```
Registry.put("Mar24MWFLD",
    Mar24MWFLDStudent)
```

```
Student shubham = Registry.get("Mar24MWFLD").copy()
```

```
shubham.name = "_____"
```

```
shubham.age = _____
```

### Step 1

⇒ Identify the class for which we want to use prototype.

⇒ Implement copy() / clone() method.

Note: All the child classes must implement the clone() method.

### Step 2

⇒ Create a Registry to store prototype objects.

### Step 3

⇒ When we want to create an object, create a copy of prototype & just set the attrs which are different for each object.

————— \* —————  
Implementation.

10:50 Pm.