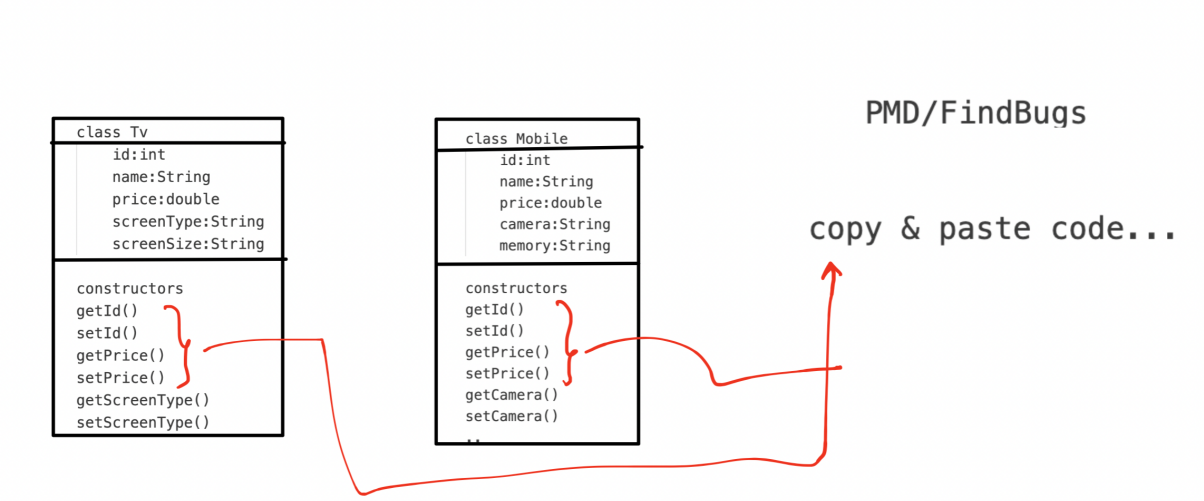
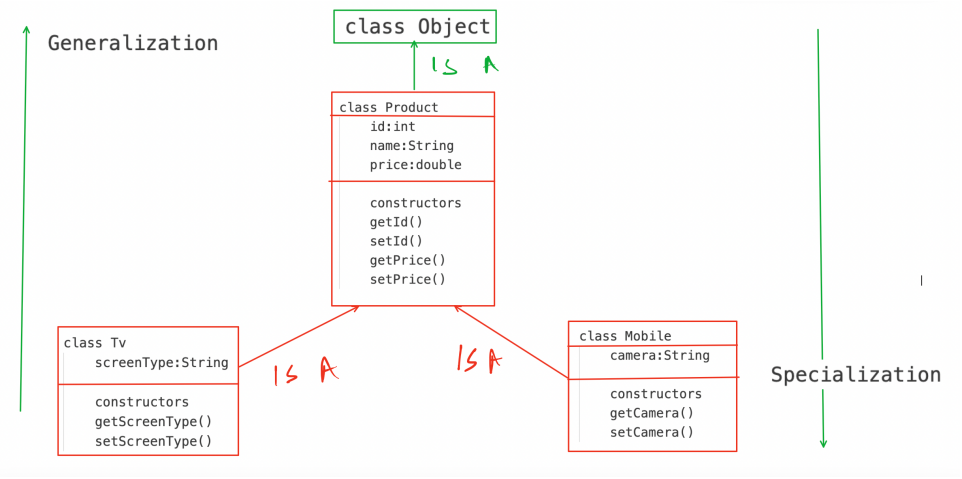
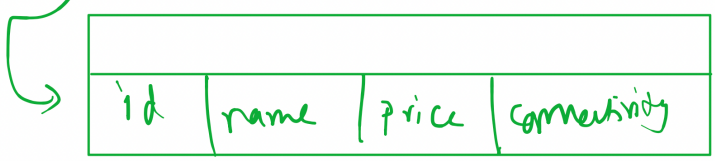
Problem: Copy and Paste

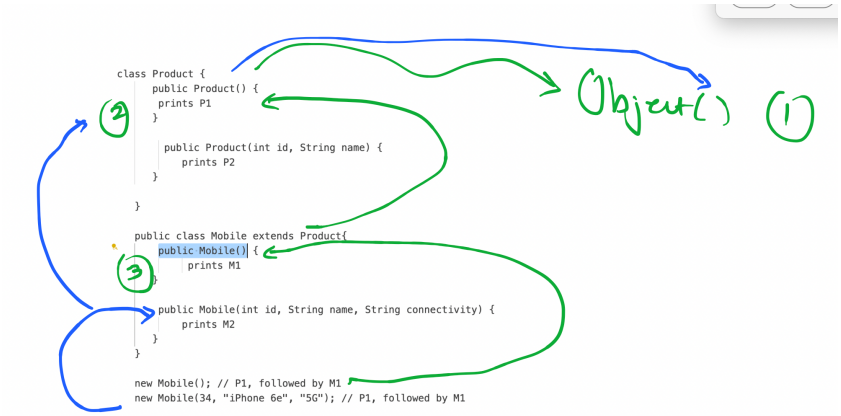


Solution: Code reusability using Generalization and Specialization relationship
Generalization and Specialization is done using **Inheritance** in programming




```
class Product {
    private int id;
    private String name;
    private double price;
}

public class Mobile extends Product{
    private String connectivity;
}

new Mobile(); // check PDF
```



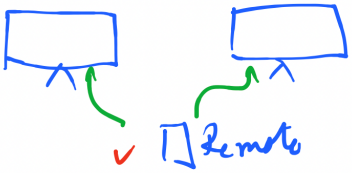



```
class Product {
    public Product() {
     prints P1
    }

    public Product(int id, String name) {
        prints P2
    }

}

public class Mobile extends Product{
    public Mobile() {
        prints M1
    }

    public Mobile(int id, String name, String connectivity) {
        super(id, name);
        prints M2
    }
}

new Mobile(); // P1, followed by M1
new Mobile(34, "iPhone 6e", "5G"); // P1, followed by M1
```

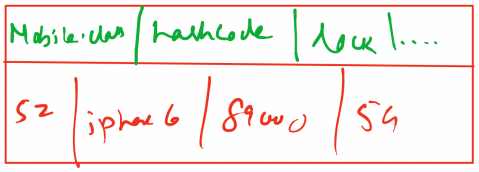
P2

```
class Product {
    public String getName() {
        return name;
    }
    public double getPrice() {
        return 0;
    }

}

public class Mobile extends Product{
    public String getConnectivity() {
        return "3G";
    }
    public double getPrice() { // override
        return 50,000;
    }
}
```

```
Product p = new Mobile(1, "Samsung Fold");
p.getName(); // valid
p.getPrice(); // dynamic binding --> 50,000.00
p.getConnectivity(); //
```



```
products[0] = new Mobile( id: 52,  name: "iPhone 6",  price: 89000.00,  connectivity: "5G");
```




```
products[0] = new Mobile( id: 52,  name: "iPhone 6",  price: 89000.00,  connectivity: "5G");
```

```
Method[] methods = p.getClass().getMethods();
```

```
for(Product p : products) {
    Method[] methods = p.getClass().getMethods();
    for(Method m : methods) {
        if(m.getName().startsWith("get")) {
            // need to invoke them
        }
    }
}
```