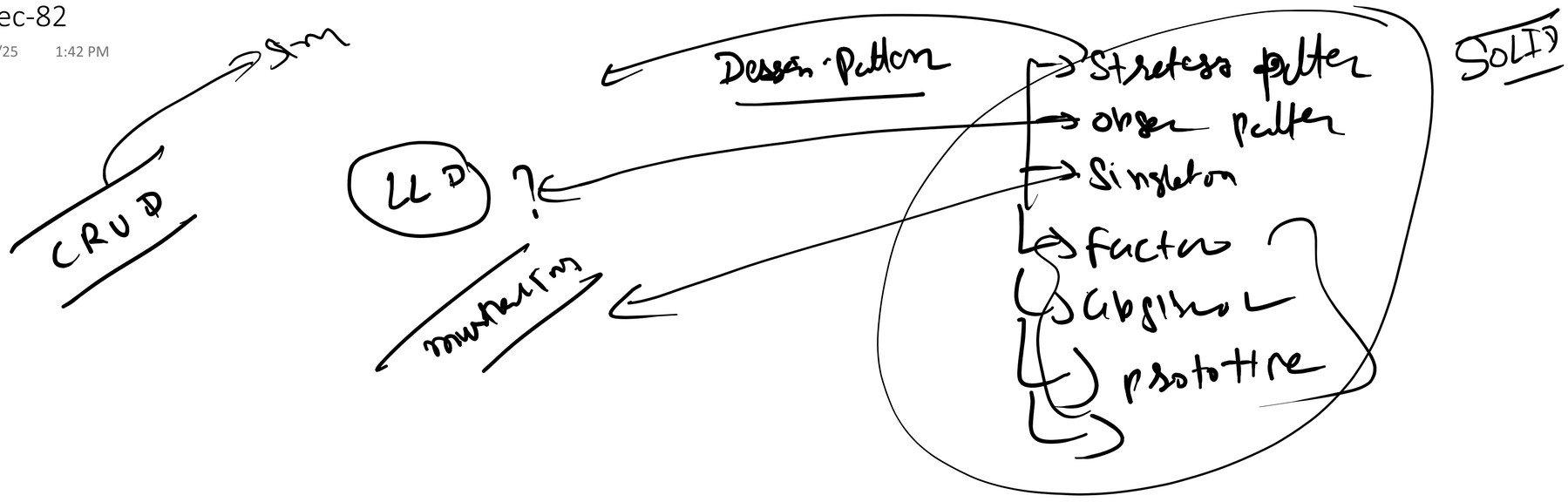
Lec-82

25/06/25 1:42 PM




## SOLID Principles

- **S** Single responsibility principle
- **O** Open/closed principle
- **L** Liskov substitution principle
- **I** Interface segregation principle
- **D** Dependency inversion principle

S. O. L. I. D

```
public class Pen {
    String color;
    int price;
}
```

```
public class Invoice {
    Pen p;

    public Invoice(Pen p) {
        // TODO Auto-generated constructor stub
        this.p = p;
    }

    public int CrateInvoice() {
        return (int) ((p.price * 1.18) + 2);
    }

    public void PrintInvoice() {
        System.out.println("Invoice Print kra");
    }
    public void SendMail() {
        System.out.println("Mail pe sendz kra");
    }

}
```

```
public class Client {
    public static void main(String[] args) {
        Pen p = new Pen();
        Invoice i = new Invoice(p);
        System.out.println(i.CrateInvoice());
        i.PrintInvoice();
        i.SendMail();
    }
}
```

if B

Open Extension Closed for modification

S.O.L

if B subtype of class A

List<Integer> ll = new ArrayList<>()
= new LinkedList<>()

Interface multifunDevice {
void print()
void Scan()
void fax()
}

BasicPrinter implements multifunDevice {
print()
Scan()
fax()
}