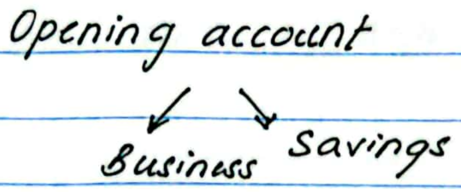
21st Sept



```
Process payment
 if (business...)
     ....
else if (savings)
     ....
....
```

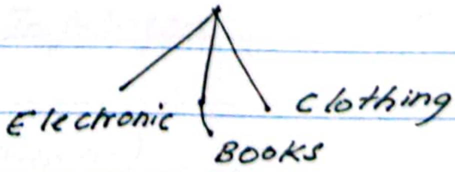
- In case, another type is introduced, need to modify everything.

Factory Design Pattern

When? Creational, similar kind of object creation

How? Abstract creation process

example; Amazon products



If there is functional difference between each type, then we require factory design pattern.

If no change in creation/functionalites between the types, a simple property type will do.

Parts of Factory Design Pattern:

① interface

<<ILogger>>
log()

② factory class

Switch - log level
→ creates Logger of log level type
→ returns the logger object

③ Actual Logger classes which implement ILogger

→ DebugLogger implements ILogger
→ ErrorLogger " "
→ InfoLogger " "

④ Usage: In Main()
Create using Logger Factory

```
Logger debugLogger = LoggerFactory.createLogger
                                (LogLevel.Debug)
```

SOLID

Single Responsibility - Creation

OCP - Easy to extend without modifying

DIP - High level module no longer relies on low lvl

Problem in simple factory.

- Factory still violates OCP.
- As it has a switch statement to find which object to create & create it

Factory Method Design Pattern.

- Factory itself made into interface

① Interfaces

«ILoggerFactory»　　　　«ILogger»

② Factories

- DebugFactory implements ILoggerFactory
- InfoFactory " "
- Error Factory " "

③ Logger classes —same as before

→ DebugLogger implements ILogger

→ ErrorLogger implements ILogger

→ InfoLogger implements ILogger

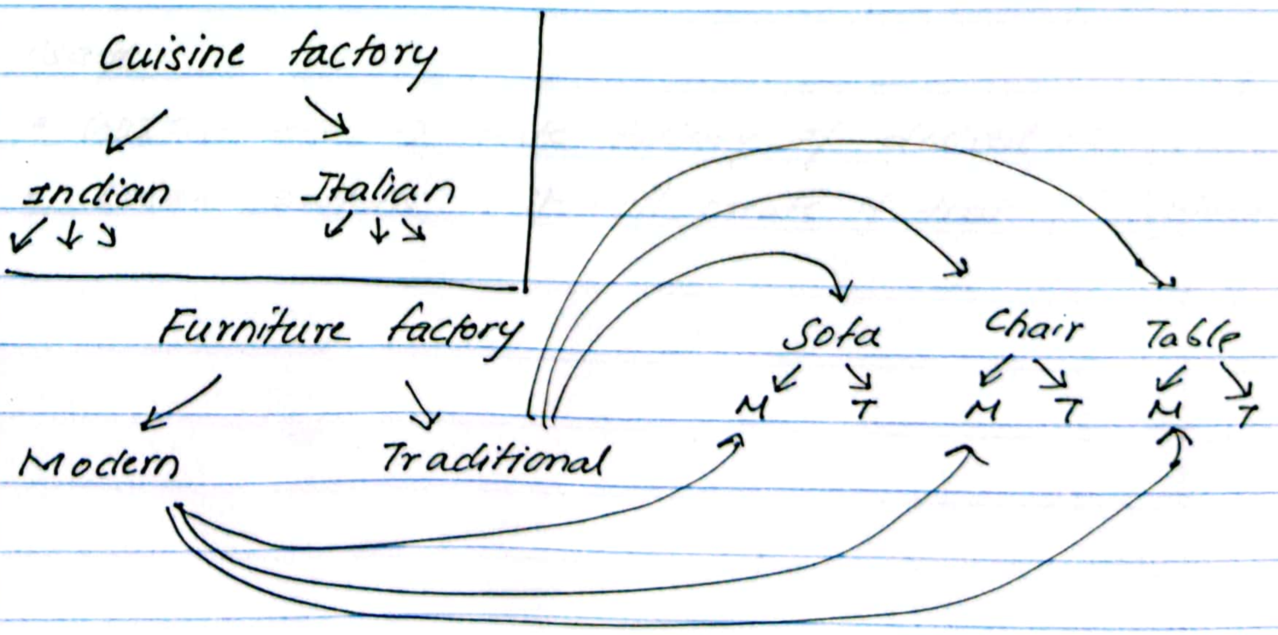
④ Usage :

Create LoggerFactory & use that to create loggers.

```
ILoggerFactory InfoFactory1 = new InfoFactory();
ILogger infoLogger = [illegible] InfoFactory1.
                                        CreateLogger().
```

.... So on - for all types

<u>Abstract Factory Design Pattern.</u>



① Interfaces :

IFactory
↳ create button
↳ create text

IButton
↳ press

IText
↳ setText

② classes :

(a) MacFactory implements IFactory → create macbutton,text
Win Factory implements IFactory → create winbutton,tt

(b) Mac Button implements IButton
Win Button implements IButton

Mac Text implements IText
Win Text implements IText

③ Usage :

- GUIFactory → create factory of desired os.
- Create buttons (it will create of that os)

Eg
MacFactor
MacButton