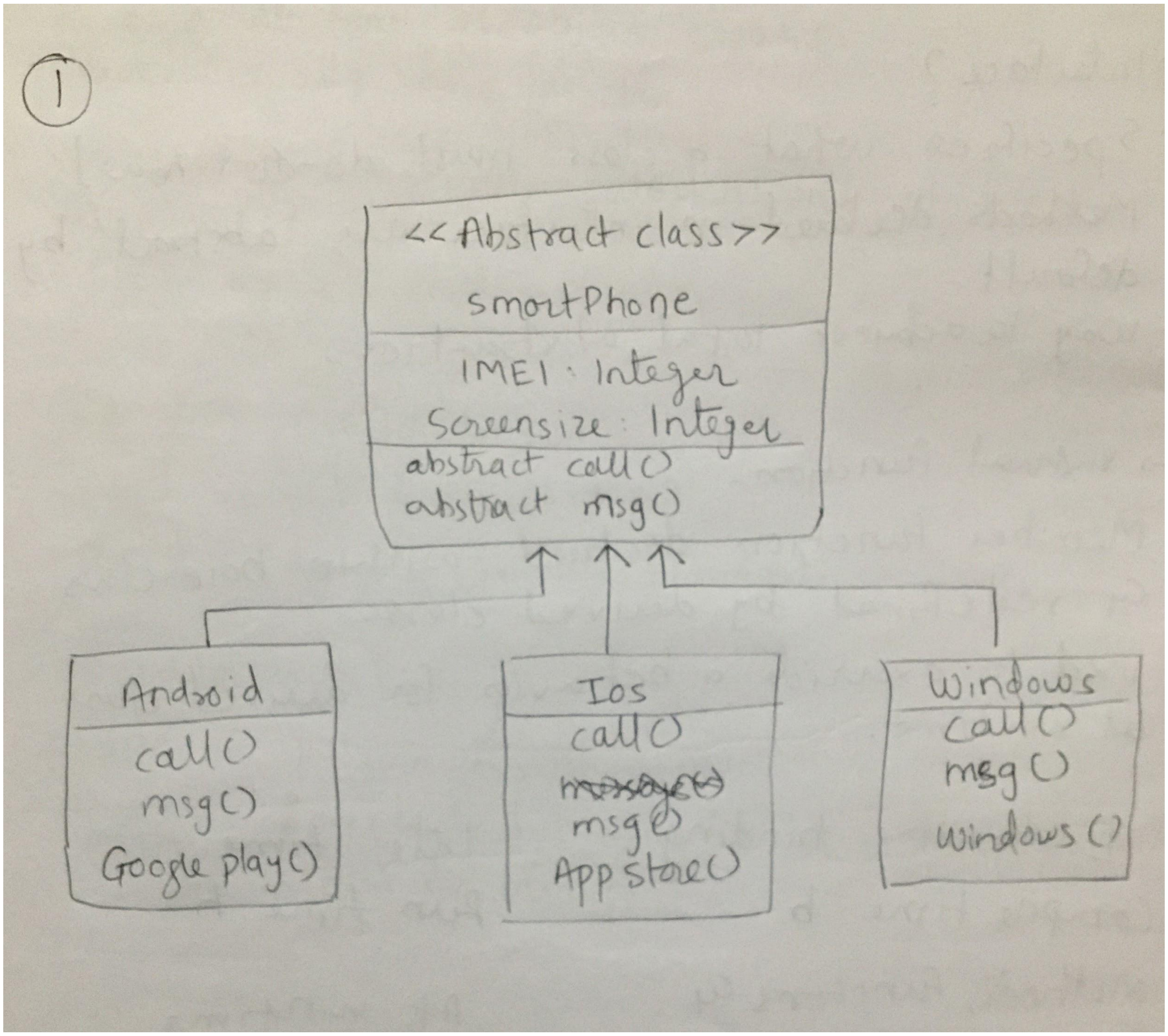
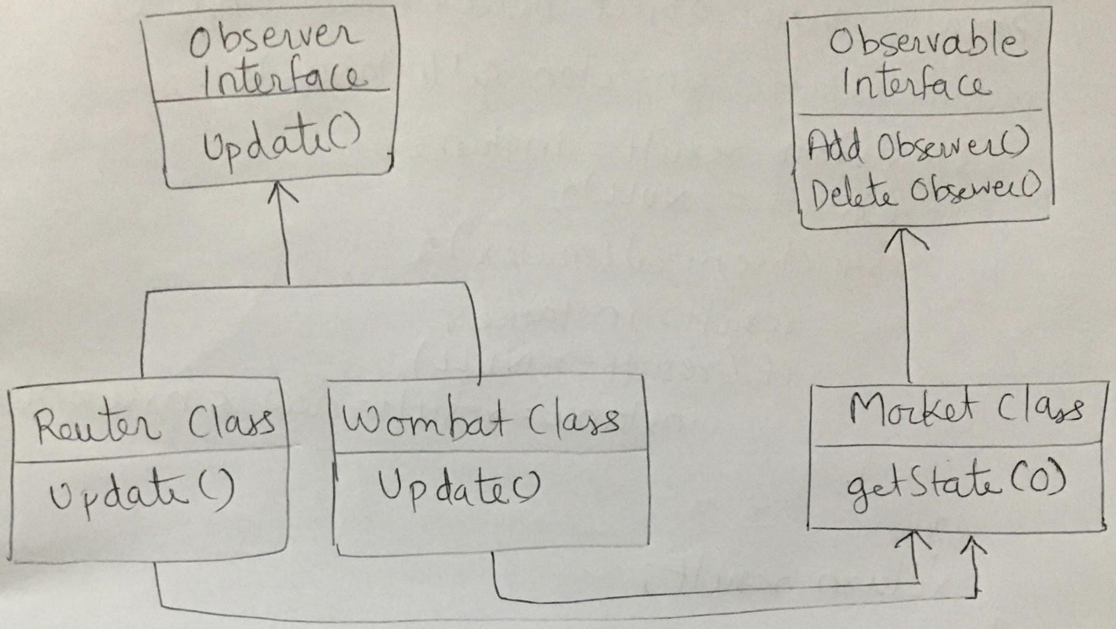
Parth Modhvadia ***(INT633)***



②



③ Simplest design pattern.

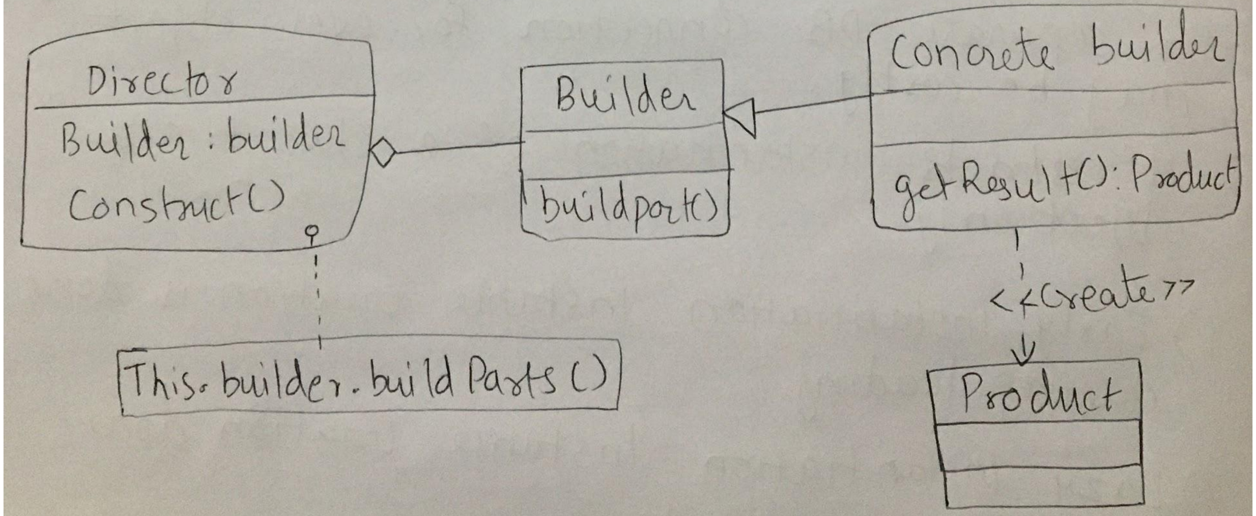
- Used when we need one instance of our class, which is shared by multiple objects, as creating a separate DB connection for every object may be costly.
- It restricts instantiation of a class to one object only.

1) Early Instantiation : Instance creation is done during loading.

2) Lazy Instantiation : Instance creation done only when required.

```
public class singleton{
   private static volatile singleton instance;
   private static object mutex = new Object();
  public static singleton getInstance(){
     singleton result = instance;
     if (result ==NULL){
         synchronized(mutex){
              result= instance;
              if(result= =NULL)
                  instance=result= new singleton();
         }
     }
     return result;
   }
}
```

4


Director
Builder : builder
Construct()
This. builder. build Parts ()
Builder
buildpart()
Concrete builder
getResult(): Product
<<Create>>
Product