Simple Factory

What pblm simple factory solves ?

Multiple types can be initiated and the choice is based on some simple criteria.

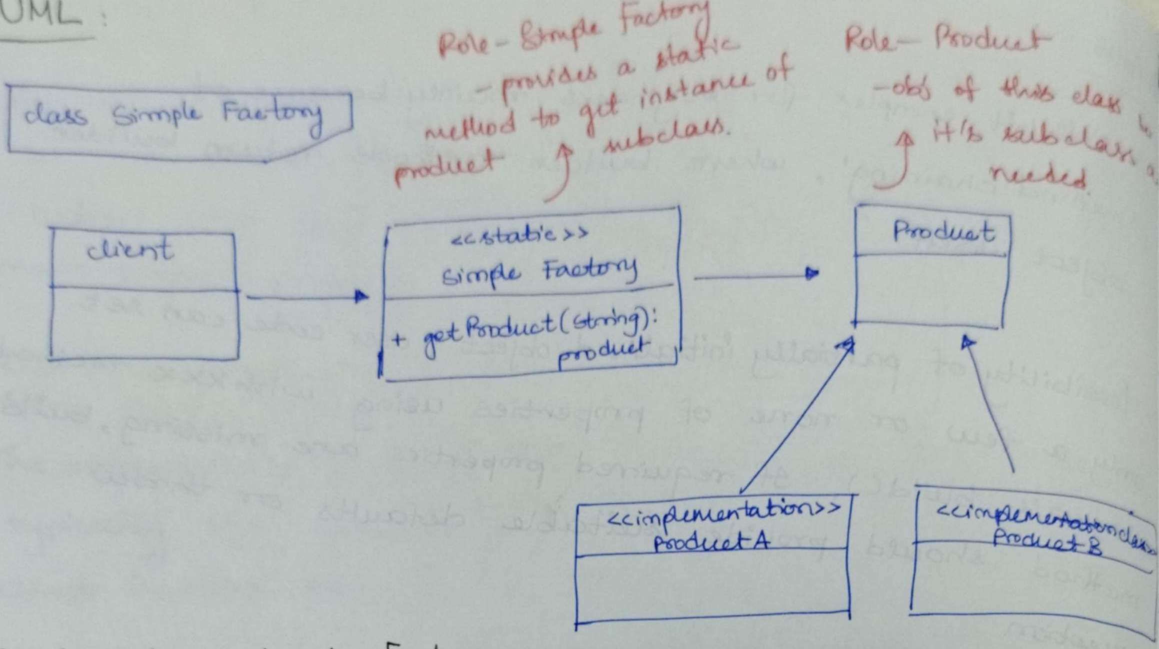
```
if ( Key. equalsIgnoreCase ("pudding")) {
    // create pudding object
} else if( key. equalsIgnoreCase ("cake")) {
    // create cake object
}
```

What is a Simple Factory ?

→ Here we simply move the instantiation logic to a seperate class and most commonly to a static method of this class.

→ Some do not consider simple factory to be a "design pattern", as its simply a method that encapsulate object instantiation. Nothing complex goes on in that method.

UML :



class Simple Factory
Role - Simple Factory - provides a static method to get instance of product ↑ subclass.
Role - Product - obj of this class & it's subclass needed
client
<<static>> Simple Factory
+ getProduct(String): product
Product
<<implementation>> Product A
<<implementation class>> Product B

Implement a Simple Factory

→ We start by creating a seperate class for our simple factory
  - Add a method which returns a desired object instance
    ◦ This method is typically static and will acept some argument to decide which class to instantiate
    ◦ You can also provide additional arguments which will be used to instantiate objects.

Implementation Considerations :-

→ Simple Factory can be just a method in existing class. Adding a seperate class however allows other parts of your code to use simple factory more easily

→ Simple Factory itself doesn't need any state tracking so it's best to keep this as static method.

Design Considerations

→ Simple factory will in turn use other design patterns like builder to construct objects.

→ In case you want to specialise your simple factory in sub-class you need factory method design pattern instead

Example :

→ The java.text.NumberFormat class has getInstance method, which is an example of simple factory

Pitfalls :

→ The criteria used by simple factory to decide which object to instantiate can get more convoluted / complex over time. If you find yourself in such situation then use factory method design pattern.