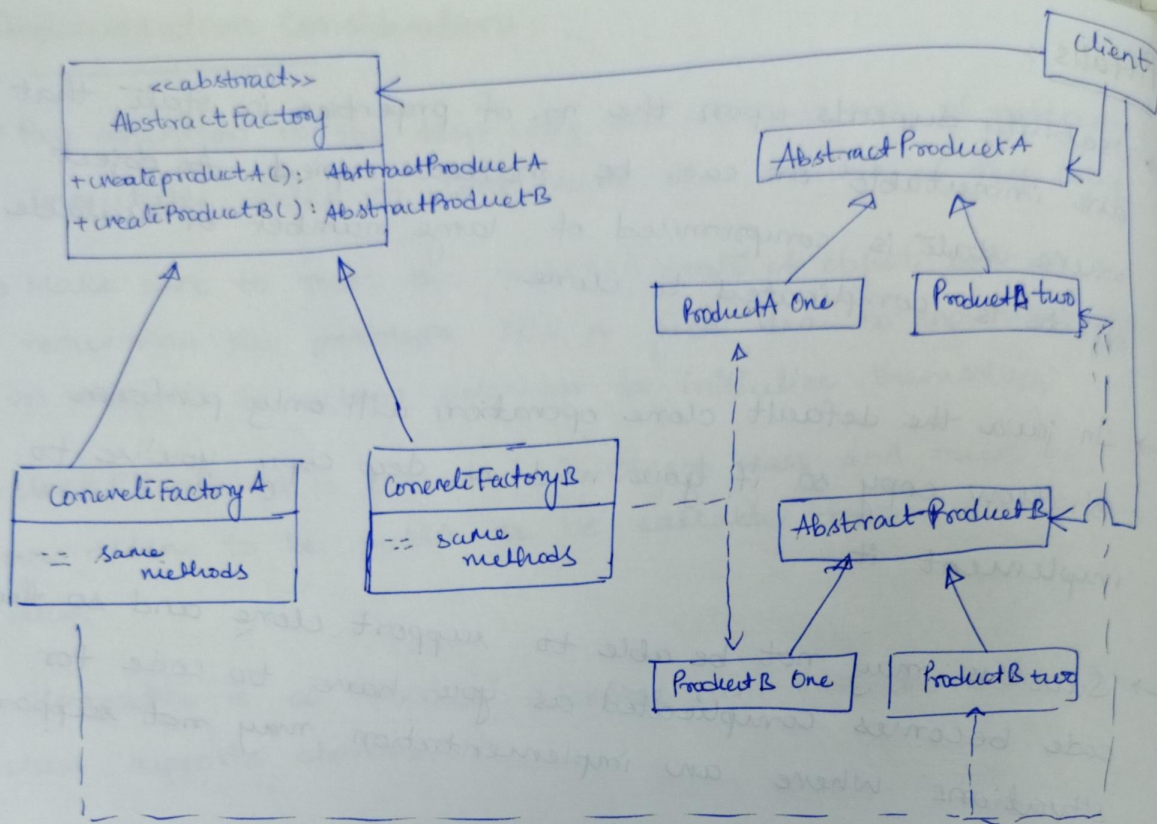


Abstract Factory :-

What is Abstract Factory?

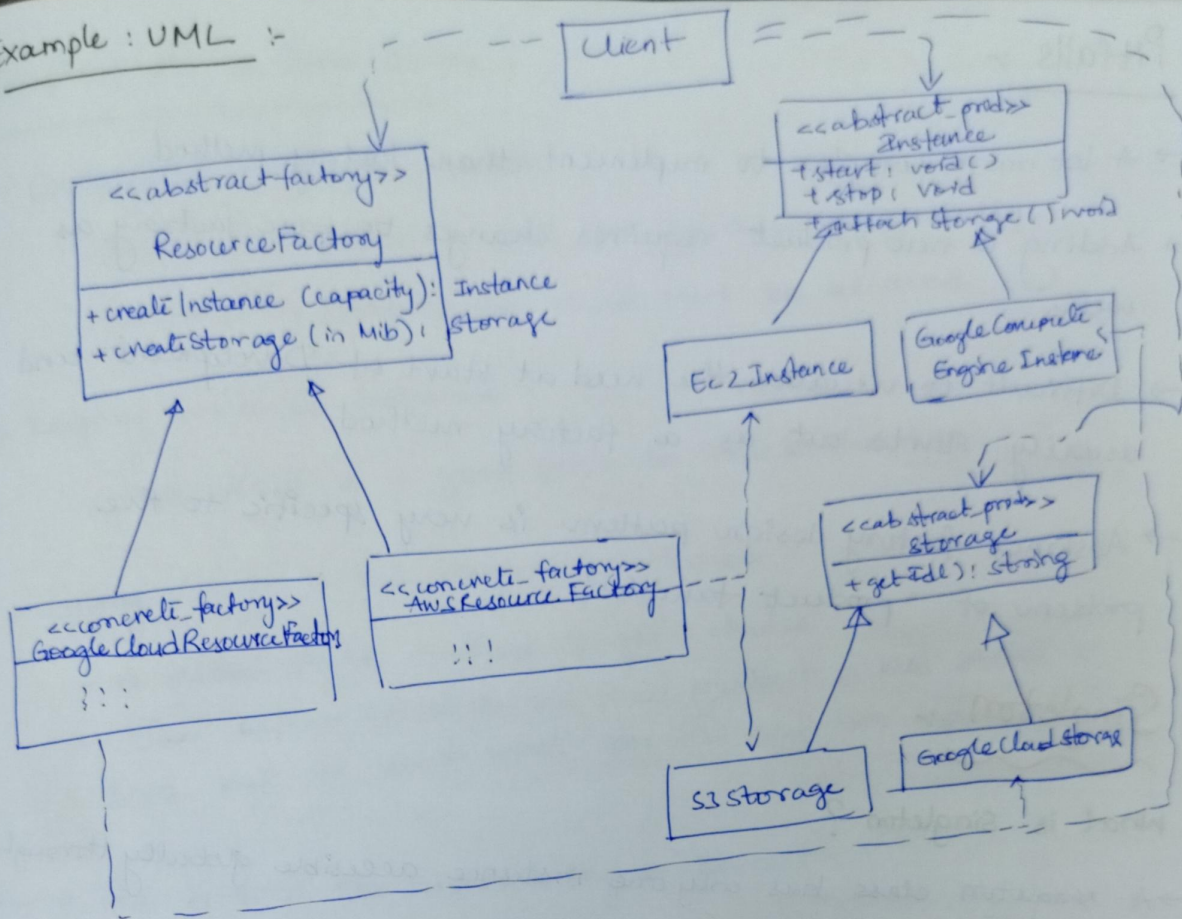
- Abstract factory is used when we have two or more objects which work together, forming a kit or set and there can be multiple kits or sets that can be created by client code.
- So we separate client code from concrete objects forming such a set and also from the code which creates these sets.



Implementation :-

- We start by studying the product "sets"
 - create abstract factory as an abstract class or an interface
 - create Abstract factory defines abstract methods for creating products.
 - Provide concrete implementation of factory for each set of products.
- Abstract Factory makes use of factory method pattern. You can think of abstract factory as an object with multiple factory methods.....

Example : UML :-



Implementation Considerations

- Factories can be implemented as singletons, we typically even need only one instance of it anyway. ✗
- Adding a new product type requires changes to the base factory as well as all implementations of factory.
- We provide the client code with concrete factory so that it can create objects

Design Considerations

- When we want to constrain object creations so that they all work together then abstract factory is good design pattern.
- Abstract Factory uses factory method pattern
- If objects are expensive to create then you can transparently switch factory implementation to use prototype design pattern to create objects

Pitfalls :-

- A lot more complex to implement than factory method.
- Adding a new product requires changes to base factory as well.
- Difficult to visualize the need at start of Development and usually starts out as a factory method.
- Abstract factory design pattern is very specific to the problem of "product families".