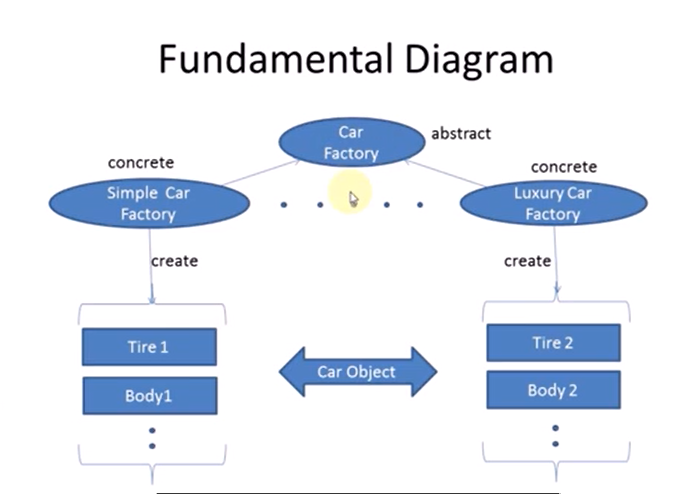
**Abstract Factory Design Pattern:**

AFDP Defines an abstract class for creating families of related objects but without specifying their concrete sub-class



**why/when to use AFDP:**

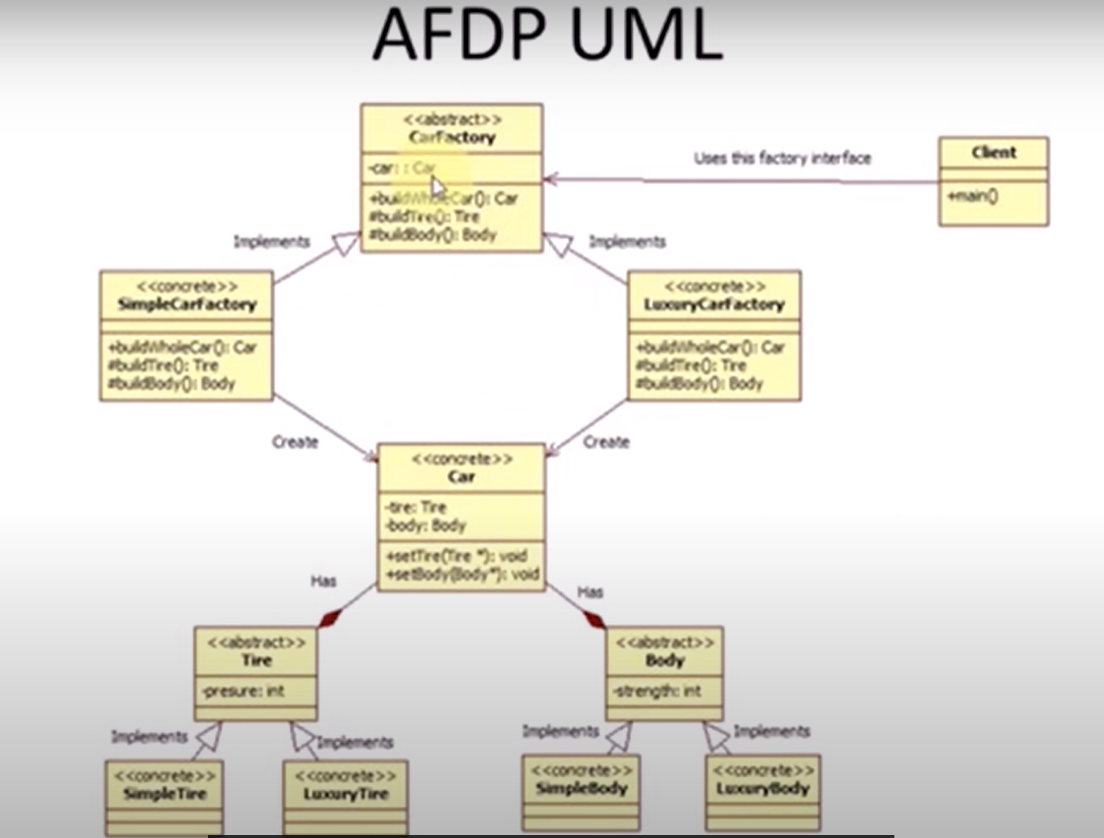
1. you need system to be independent of how objects are created,

composed and represented.

2. show interface not implementation.

3. system need to be configured with one of the multiple family

of objects.



**Usage examples:** The Abstract Factory pattern is pretty common in C++ code. Many frameworks and libraries use it to provide a way to extend and customize their standard components.

**Identification:** The pattern is easy to recognize by methods, which return a factory object. Then, the factory is used for creating specific sub-components.

**More Examples at:**  
https://refactoring.guru/design-patterns/abstract-factory/cpp/example#lang-features