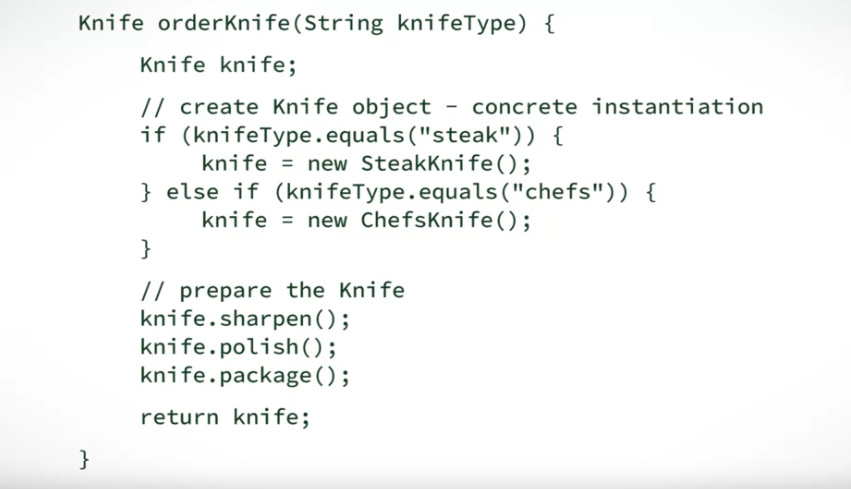
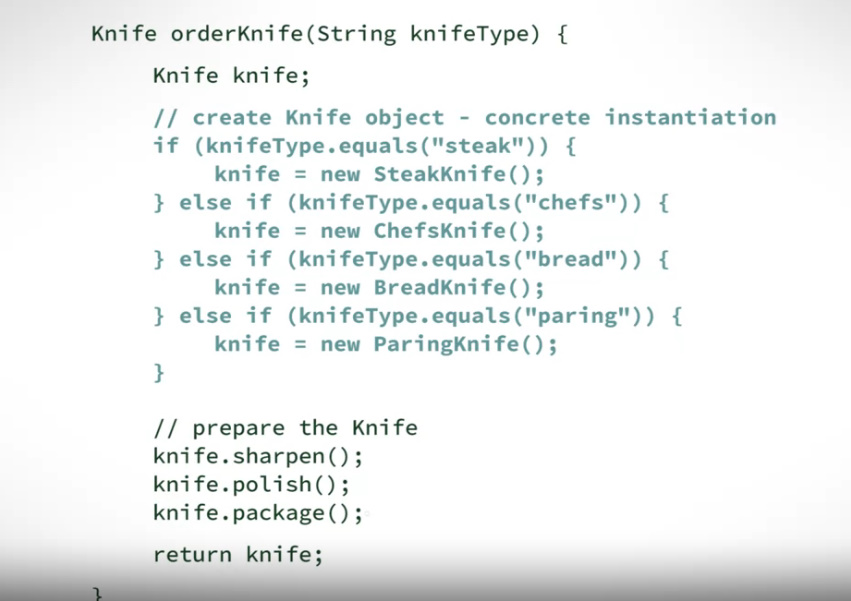
**Creational Factory Pattern**

* Factory are rolul de a crea un object.
* De ex, fie o clasa Knife, si diferite subclase, ca ChefsKnife, SteakKnife
* Acum, cream o metoda care, in dependenta de alegerea facuta, creaza un anumit obiect, gen ori ChefsKnife sau SteackKnife, si returneaza un object de tip Knife,

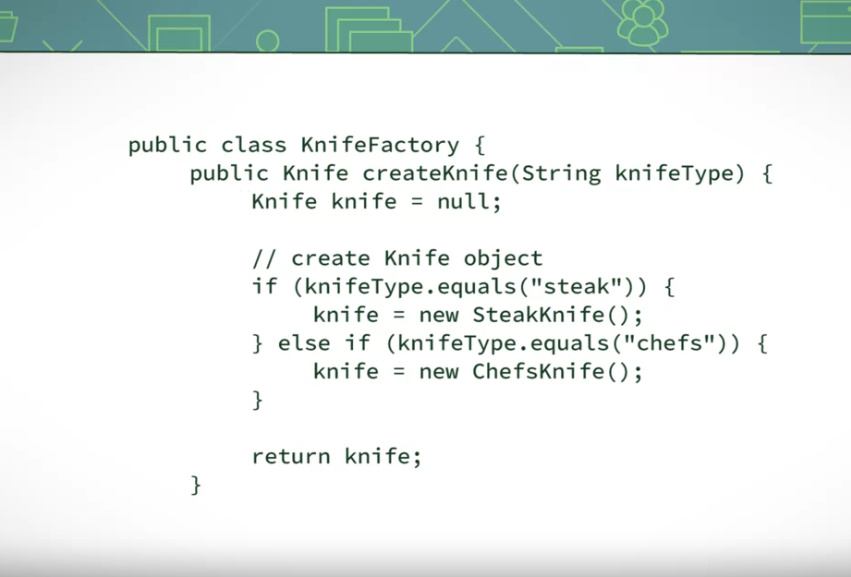


1. Cream metoda ce va returna un object de tip Knife, oferim un parametru ca sa putem alege tipul la objec
2. Cream o variabila locala de tip Knife, ce sa pastreze referinta la obiectul creat
3. Cream obiectul potrivit in dependenta de parametrul oferit prin Concrete Instatiation( cu new). Putem executa si metode de ale sale pentru a mai crea ceva in el.
4. returnam obiectul creat

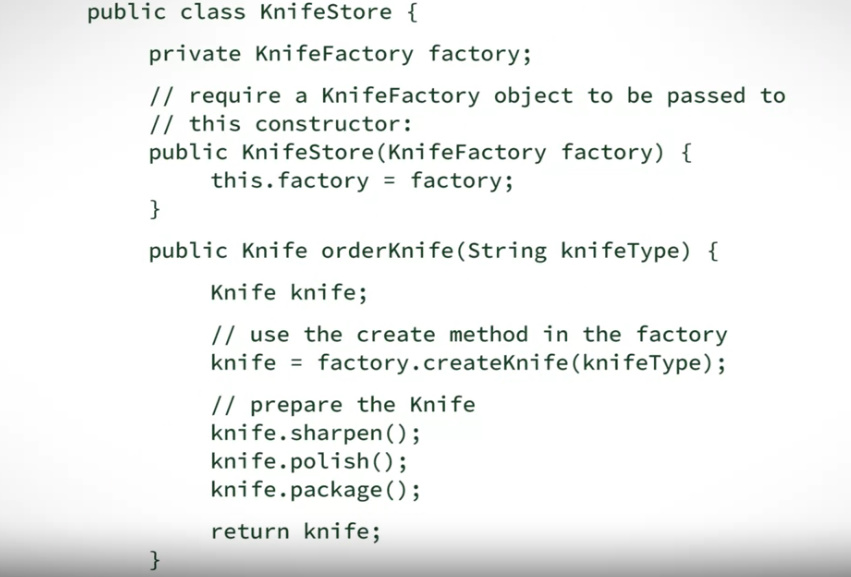
* Totusi, daca cream mai multe tipuri de Knife, metoda va creste mult



* Metoda data o punem intr-o alta clasa, deobicei cu sufixul Factory, ca KnifeFactory



* **Factory Object** – instanta unei clase ce are o metoda pentru a crea un product obiect.
* **KnifeFactory** poate fi utilizat ca field in alte clase apoi, ca gen in Store:



KnifeFactory este un client al KnifeStore

**Factory Design Pattern prevede ca cream doar o clasa ce sa fie Factory, si ea creaza deja diferite Obiecte in ea, fie intr-o metoda, fie in mai multe, dar avem doar o instanta la Fctory, fara subclasele sau interfete.**