```java
// Abstract Product
abstract class Pizza {
    public abstract void prepare();
    public abstract void bake();
    public abstract void cut();
    public abstract void box();
}

// Concrete Products
class CheesePizza extends Pizza {
    public void prepare() {
        System.out.println("Preparing Cheese Pizza");
    }

    public void bake() {
        System.out.println("Baking Cheese Pizza");
    }

    public void cut() {
        System.out.println("Cutting Cheese Pizza");
    }

    public void box() {
        System.out.println("Boxing Cheese Pizza");
    }
}

class PepperoniPizza extends Pizza {
    public void prepare() {
        System.out.println("Preparing Pepperoni Pizza");
    }

    public void bake() {
        System.out.println("Baking Pepperoni Pizza");
    }

    public void cut() {
        System.out.println("Cutting Pepperoni Pizza");
    }

    public void box() {
        System.out.println("Boxing Pepperoni Pizza");
    }
}
```

```
// Factory interface
interface PizzaFactory {
    Pizza createPizza(String type);
}

// Concrete Factory: New York Pizza Factory
class NYPizzaFactory implements PizzaFactory {
    public Pizza createPizza(String type) {
        if (type.equals("cheese")) {
            return new CheesePizza();
        } else if (type.equals("pepperoni")) {
            return new PepperoniPizza();
        }
        return null;
    }
}

// Client code to test Factory Method Pattern
public class PizzaStore {
    public static void main(String[] args) {
        PizzaFactory nyFactory = new NYPizzaFactory();
        Pizza pizza = nyFactory.createPizza("cheese");
        pizza.prepare();
        pizza.bake();
        pizza.cut();
        pizza.box();
    }
}
```