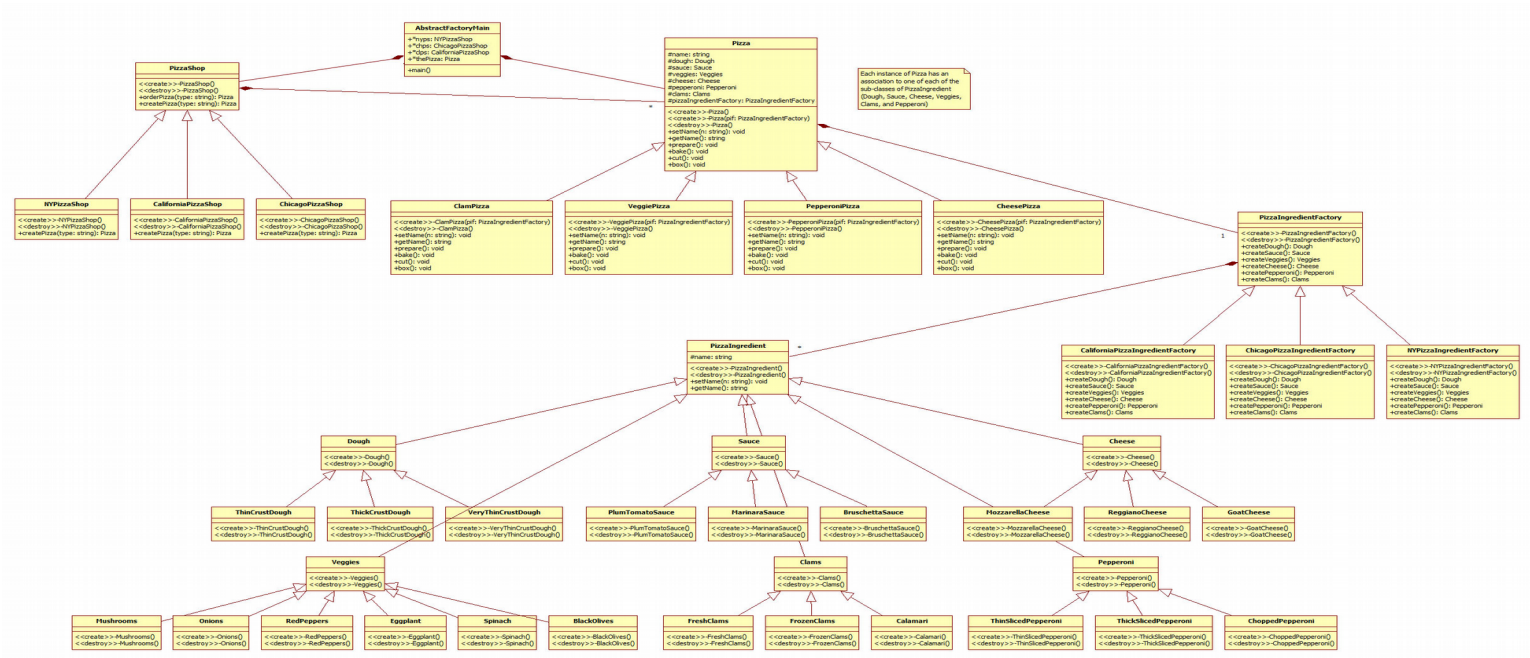
Abstract Factory Design Pattern

GoF Statement:	**Provides an interface for creating families of related or dependent objects without specifying their concrete classes.**
Category:	**Creational**
UML Diagram:	



Description of the Demonstration:

There are three subclasses of PizzaShop (NYPizzaShop, ChicagoPizzaShop, and CaliforniaPizzaShop) each of which overrides the virtual function PizzaShop::createPizza. There are four types or sub-classes of Pizza; CheesePizza, ClamPizza, VeggiePizza, and PepperoniPizza. Each type of pizza consists of up to six PizzaIngredients: Dough, Sauce, Cheese, Veggies, Clams and Pepperoni. But, each regional pizza shop uses different versions (sub-classes) of these five ingredients. To handle creation of pizzas specific to the region an instance of Pizza, whether CheesePizza, ClamPizza, VeggiePizza or PepperoniPizza is given a pointer to an instance of the appropriate regional sub-class of the parent PizzaIngredientFactory which is the Abstract Factory. This Abstract Factory class defines the standard interface functions createDough, createSauce, createCheese, createVeggies, createClams, and createPepperoni. It is the sub-class of PizzaIngredientFactory that handles the actual creation of the concrete Pizza class with the correct regional style ingredients appropriate to its' regional PizzaShop.