

# Learning Objectives

- Understand class methods
- Understand how constructor methods work
- Prepare them to understand composition of methods into convenience methods

## Repository

Download and use the [PizzaRestaurant](#) starter package for this exercise.

## Instructions

You'll be modelling a pizza restaurant. Specifically, the kitchen of a delivery pizza restaurant. You will continue working on this assignment tomorrow.

We won't expose the chefs, the oven, or any internal workings. All we care about is that we can send a message to the kitchen asking them to make a pizza, and they'll return us a pizza.

Start by defining a pizza object. Let's give it a couple methods: one that returns a value for its size (small, medium, or large) and another that gives its toppings (an array of strings, like “ham, pineapple, cheese”). Make an `enum` type to represent the size.

Don't create any methods to set these values directly (after all, it's difficult to change a pizza's size and toppings after it's baked). You'll have to set these values yourself, somehow. If you imagine the process of baking a pizza as creating an object for use in a program, that should suggest that you should be setting these values when the object is first created.

Create an `init` method that will take a size and an array of toppings, and have that method set the internal values of the object when it is initialized.

In `main.m`, take the text input by a user, and use it to ask the kitchen to make a pizza. If the user inputs `small ham pineapple cheese`, that should generate a `Pizza` object with a `PizzaSize` of `Small`, and a toppings array containing `ham`, `pineapple`, and `cheese`.

## Stretch Goals

- Create a couple of constructor method on the `Pizza` class to make common size and topping combinations. For example `+(Pizza *)largePepperoni`, or `+(Pizza *)meatLoversWithSize:(PizzaSize)size`.
- Then, allow users to use the constructor methods you've created to request common pizzas. A user inputting `pepperoni` could call the `+(Pizza *)largePepperoni` class method, for example. But notice you really don't want to call these class methods directly from `main.m`. Let the Kitchen handle making pizzas. Otherwise, the next thing you know `main.m` will be helping itself to the pop fridge too! :)