Lesson 6 Homework: Inventory App Using Classes

# Ruby Foundations Review

If you did not yet complete the last question from the Review we did yesterday, please complete it now:

Write code to create a new instance of a Vehicle object and make it honk.  
class Vehicle  
 def initialize(color, type)  
 @color = color  
 @type = type # car, truck, motorcycle, scooter, van  
 end  
 def honk  
 puts "Beep!"  
 end  
end

# Inventory App

Create a program that:

1. Lists several items
2. Allows a user to select an item and view the inventory count, change the number of the inventory, delete the item altogether, or even change the name.
3. Create a new item and give the item an inventory count, and have this item now show up in the list of items you can select.
4. Loops through until the user does not want to edit the list anymore.

# Bonus! Get started on exercism

## What is exercism?

[Exercism](http://exercism.io/) is an excellent web application that helps you develop deep programming expertise through both practice and crowdsourced mentorship. It has exercises in many languages, including Ruby.

## How to Install

### Step 1: Install the CLI

Find links to how to install on your system type [here](http://help.exercism.io/installing-the-cli.html). Find your system, then click through to find the link to the instructions for **Installing the CLI for <system>**. **Mac** users should definitely use **homebrew**.

### Step 2: Configure the CLI

Find instructions [here](http://help.exercism.io/configuring-the-cli.html).

### Step 3: Fetch your first exercism

Find the full instructions [here](http://help.exercism.io/fetching-exercises.html), but basically type “exercism fetch ruby” in your command line.

The first exercism has step-by-step instructions listed in the files. This will probably be your first exposure to testing, so don’t worry if it seems confusing at first. Grab one of your instructors if you need help getting started.

# Extra Bonus!

Make a simple app that counts to 100, but does the following things:

* if the number is divisible by 3, print Fizz
* if the number is divisible by 5, print Buzz
* if both, print FizzBuzz
* otherwise print the number