**Store & Products**

Objectives:

* Practice creating classes
* Practice associations between classes
* Practice modularizing code

Start by creating a Store class that has 2 attributes: a name and a list of products. The name must be provided upon creation, but the products list should be empty.

Next, create a Product class that has 3 attributes: a name, a price, and a category. All of these should be provided upon creation.

Let's give some methods to our Product class:

* **update\_price(self, percent\_change, is\_increased)** - updates the product's price. If is\_increased is True, the price should increase by the percent\_change provided. If False, the price should decrease by the percent\_change provided.
* **print\_info(self)** - print the name of the product, its category, and its price.

Let's also give some methods to our Store class:

* **add\_product(self, new\_product)** - takes a product and adds it to the store
* **sell\_product(self, id)** - remove the product from the store's list of products given the id and print its info.
* **inflation(self, percent\_increase)** - increases the price of each product by the percent\_increase given
* **set\_clearance(self, category, percent\_discount)** - updates all the products matching the given category by reducing the price by the percent\_discount given
* Top of Form
* Create a Store class with 2 attributes
* Create a Product class with 3 attributes
* Add the print\_info method to the Product class
* Add the update\_price method to the Product class
* Add the add\_product method to the Store class
* Add the sell\_product method to the Store class
* Test out your classes by creating an instance of the Store and a few instances of the Product class, add those instances to the store instance, and then test out the methods.
* NINJA BONUS: Add the inflation method to the Store class
* NINJA BONUS: Add the set\_clearance method to the Store class
* NINJA BONUS: Modularize your code into 3 separate files
* SENSEI BONUS: Update the product class to give each product a unique id. Update the sell\_product method to accept the unique id.Bottom of Form