* [Object Oriented Programming](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297" \l "Object_Oriented_Programming)
  + [Abstract](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297#Abstract)
* [I. Classes](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297#I._Classes)
* [II. Objects (instances)](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297#II._Objects_(instances))
* [III. Defining a class](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297#III._Defining_a_class)
* [IV. Creating an object (instance)](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297#IV._Creating_an_object_(instance))
* [V. Attributes](http://learn.di-learning.com/courses/collection/18/course/13/section/61/chapter/297#V._Attributes)

1. Create a class called Dog.
2. In this class, create an \_\_init\_\_ method that takes two parameters : name and height. This function instantiates two attributes, which values are the parameters.
3. Create a method called bark that prints the following string “<dog\_name> goes woof!”.
4. Create a method called jump that prints the following string “<dog\_name> jumps <x> cm high!”. x is the height\*2.
5. Outside of the class, create an object called davids\_dog. His dog’s name is “Rex” and his height is 50cm.
6. Print the details of his dog (ie. name and height) and call the methods bark and jump.
7. Create an object called sarahs\_dog. Her dog’s name is “Teacup” and his height is 20cm.
8. Print the details of her dog (ie. name and height) and call the methods bark and jump.
9. Create an if statement outside of the class to check which dog is bigger. Print the name of the bigger dog.

#### Instructions

The purpose of this exercise is to create a restaurant menu. The code will allow a manager to add and delete dishes.

1. Create a python file called menu\_manager.py.
2. Create a class called MenuManager.
3. Create a method \_\_init\_\_ that instantiates an attribute called menu. The menu attributes value will be all the current dishes (**list of dictionaries**). Each dish will be stored in a dictionary where the keys are name, price, spice level and gluten index (which value is a boolean).

Here is a list of the dishes currently on the menu:

Soup – 10 – B - False

Hamburger – 15 - A - True

Salad – 18 - A - False

French Fries – 5 - C - False

Beef bourguignon– 25 - B - True

Meaning: For the spice level:

• A = not spicy,

• B = a little spicy,

• C = very spicy

The dishes provided above should be the value of the menu attribute.

1. Create a method called add\_item(name, price, spice, gluten). This method will add new dishes to the menu.
2. Create a method called update\_item(name, price, spice, gluten). This method checks whether a dish is in the menu, if the dish exists then update it. If not notify the manager that the dish is not in the menu.
3. Create a method called remove\_item(name). This method should check if the dish is in the menu, if the dish exists then delete it and print the updated dictionary. If not notify the manager that the dish is not in the menu.