

## Creating Burger class

in list DONE

DUE DATE

☐ May 22 at 12:00 AM COMPLETE 

### Description Edit

- Create initialization class with name and ingredient as arguments.
- create class variable 'name' with a local variable `name` as a default value.
- create class variable 'menu' with new instance of the menu class as a default value.
- create class variable 'order' with new instance of the order class as a default value.
- create attribute reader permission to the class variables name, menu and order.
- create a method 'print\_ingredients' with no arguments, it should call the menu method 'display\_ingredients'.
- create a method 'populate\_menu' with 'ingredient' as argument, it should loop through each key and value of the argument hash array calling the menu method add\_ingredient with the key and value as arguments in order to add the ingredients name and price to the menu class variable 'ingredients'. call this method on the class initialization.
- create a method 'add\_to\_order' with 'ingredient' and 'quantity' as arguments, it should call the 'add\_order\_ingredient' method from the instantiated order class on the order class variable and take the same arguments passed by the method 'add\_to\_order'.
- create a method 'display\_order' with no arguments, it should call the method 'display' from the instantiated order class on the order class variable.
- create a method 'clear\_order' with no arguments, it should call the 'clear\_order' method from the instantiated order class on the order class variable.
- create a method 'order\_total\_amount' with no argument, it should loop through the 'get\_order\_ingredients' method from the instantiated order class on the class variable order class variable, for each iteration it should call the get\_price method from the instantiated menu class on the menu class variable, multiplying the quantity by its unit price and add the result into a local variable called total, then return the total variable coerced to float type.
- create a method 'display\_order\_total\_amount' with no arguments, it print a custom message with the returned value of the method 'order\_total\_amount'.

### Checklist Hide checked items Delete

100%

- ☐ ~~Create initialization class with name and ingredient as arguments.~~
- ☐ ~~create class variable 'name' with a local variable name as a default value.~~

- ☐ ~~create class variable 'menu' with new instance of the menu class as a default value.~~
- ☐ ~~create class variable 'order' with new instance of the order class as a default value.~~
- ☐ ~~create attribute reader permission to the class variables name, menu and order.~~
- ☐ ~~create a method 'print\_ingredients' with no arguments, it should call the menu method 'display\_ingredients'.~~
- ☐ ~~create a method 'populate\_menu' with 'ingredient' as argument, it should loop through each key and value of the argument hash array calling the menu method add\_ingredient with the key and value as arguments in order to add the ingredients name and price to the menu class variable 'ingredients'. call this method on the class initialization.~~
- ☐ ~~create a method 'add\_to\_order' with 'ingredient' and 'quantity' as arguments, it should call the 'add\_order\_ingredient' method from the instantiated order class on the order class variable and take the same arguments passed by the method 'add\_to\_order'.~~
- ☐ ~~create a method 'display\_order' with no arguments, it should call the method 'display' from the instantiated order class on the order class variable.~~
- ☐ ~~create a method 'clear\_order' with no arguments, it should call the 'clear\_order' method from the instantiated order class on the order class variable.~~
- ☐ ~~create a method 'order\_total\_amount' with no argument, it should loop through the 'get\_order\_ingredients' method from the instantiated order class on the class variable order class variable, for each iteration it should call the get\_price method from the instantiated menu class on the menu class variable, multiplying the quantity by its unit price and add the result into a local variable called total, then return the total variable coerced to float type.~~
- ☐ ~~create a method 'display\_order\_total\_amount' with no arguments, it print a custom message with the returned value of the method 'order\_total\_amount'.~~

Add an item

 Activity

Show details