Basta Fazoolin'

You've started a position as the lead programmer for the family-style Italian restaurant *Basta Fazoolin' with My Heart*. The restaurant has been doing fantastically and seen a lot of growth lately. You've been hired to keep things organized.

If you get stuck during this project or would like to see an experienced developer work through it, click "**Get Unstuck**" to see a **project walkthrough video**.

Tasks

26/26 Complete

Mark the tasks as complete by checking them off

Making the Menus

1.

At *Basta Fazoolin' with my Heart* our motto is simple: when you're here with family, that's great! We have four different menus: brunch, early-bird, dinner, and kids.

Create a `Menu` class .

Hint

Remember to use the `class` keyword when creating a new class.

2.

Give `Menu` a constructor with the five parameters `self`, `name`, `items`, `start_time`, and `end_time`.

Hint

Remember that a constructor in Python is a method called `__init__`.

3.

Let's create our first menu: `brunch`. Brunch is served from 11am to 4pm. The following items are sold during brunch:

```
{
  'pancakes': 7.50, 'waffles': 9.00, 'burger': 11.00, 'home fries': 4.50, 'coffee': 1.50, 'espresso': 3.00, 'tea': 1.00, 'mimosa': 10.50, 'orange juice': 3.50
}
```

Hint

Create a `Menu` object and save it into the variable `brunch`. Call it with the arguments `"brunch"` and the `items` dictionary in the code block above.

For the `start_time` and `end_time`, either you can use a 24-hour clock (`11` & `16`, respectively) or the `time` object from the [datetime module](#).

4.

Let's create our second menu item `early_bird`. Early-bird Dinners are served from 3pm to 6pm. The following items are available during the early-bird menu:

```
{
  'salumeria plate': 8.00, 'salad and breadsticks (serves 2, no refills)': 14.00, 'pizza with quattro formaggi': 9.00, 'duck ragu': 17.50, 'mushroom ravioli (vegan)': 13.50, 'coffee': 1.50, 'espresso': 3.00,
}
```

5.

Let's create our third menu, `dinner`. Dinner is served from 5pm to 11pm. The following items are available for dinner:

```
{
  'crostini with eggplant caponata': 13.00, 'caesar salad': 16.00, 'pizza with quattro formaggi': 11.00, 'duck ragu': 19.50, 'mushroom ravioli (vegan)': 13.50, 'coffee': 2.00, 'espresso': 3.00,
}
```

6.

And let's create our last menu, `kids`. The kids menu is available from 11am until 9pm. The following items are available on the kids menu.

```
{
  'chicken nuggets': 6.50, 'fusilli with wild mushrooms': 12.00, 'apple juice': 3.00
}
```

7.

Give our `Menu` class a string representation method that will tell you the `name` of the menu. Also, indicate in this representation when the menu is available.

Hint

Define a string representation like this:

```
def __repr__(self):
  return self.representative_string
```

8.

Try out our string representation. If you call `print(brunch)` it should print out something like the following:

```
brunch menu available from 11am to 4pm
```

9.

Give `Menu` a method `.calculate_bill()` that has two parameters: `self`, and `purchased_items`, a list of the names of purchased items.

Have `calculate_bill` return the total price of a purchase consisting of all the items in `purchased_items`.

10.

Test out `Menu.calculate_bill()`. We have a breakfast order for one order of pancakes, one order of home fries, and one coffee. Pass that into `brunch.calculate_bill()` and print out the price.

Hint

If you print out

```
brunch.calculate_bill(['pancakes', 'home fries', 'coffee'])
```

You should get the value `13.5`.

11.

What about an early-bird purchase? Our last guests ordered the salumeria plate and the vegan mushroom ravioli. Calculate the bill with `.calculate_bill()`.

Hint

This purchase should result in a bill of `21.5`.

Creating the Franchises

12.

Basta Fazoolin' with my Heart has seen tremendous success with the family market, which is fantastic because when you're at *Basta Fazoolin' with my Heart* with family, that's great!

We've decided to create more than one restaurant to offer our fantastic menus, services, and ambience around the country.

First, let's create a `Franchise` class.

13.

Give the `Franchise` class a constructor. Take in an `address`, and assign it to `self.address`. Also take in a list of `menus` and assign it to `self.menus`.

14.

Let's create our first two franchises! Our flagship store is located at `"1232 West End Road"` and our new installment is located at `"12 East Mulberry Street"`. Pass in all four menus along with these addresses to define `flagship_store` and `new_installment`.

15.

Give our `Franchise`s a string representation so that we'll be able to tell them apart. If we print out a `Franchise` it should tell us the address of the restaurant.

Hint

Define a `.__repr__()` method on your `Franchise` class to give it a string representation.

16.
Let's tell our customers what they can order!
Give `Franchise` an `.available_menus()` method that takes in a `time` parameter and returns a list of the `Menu` objects that are available at that time.
Hint
Write a method `.available_menus()` that takes in two parameters, `self` and `time`. It should iterate through all of `self.menus` and return only those menus that are available between `start_time` and `end_time`.
17.
Let's test out our `.available_menus()` method! Call it with 12 noon as an argument and print out the results.
Hint
It should return both our `brunch` menu and our `kids` menu. Lucky we already gave them string representations!
18.
Let's do another test! See what is printed if we call `.available_menus()` with 5pm as an argument and print out the results.
Hint
You should get a list consisting of the early-bird, the dinner, and the kids menus.

Creating Businesses!

19.
Since we've been so successful building out a branded chain of restaurants, we've decided to diversify. We're going to create a restaurant that sells arepas!

First let's define a `Business` class.
20.
Give `Business` a constructor. A `Business` needs a `name` and a list of `franchises`.
21.
Let's create our first `Business`. The name is `"Basta Fazoolin' with my Heart"` and the two franchises are `flagship_store` and `new_installment`.
22.
Before we create our new business, we'll need a `Franchise` and before our `Franchise` we'll need a menu. The items for our *Take a' Arepa* available from 10am until 8pm are the following:

```
{
  'arepa pabellon': 7.00, 'pernil arepa': 8.50, 'guayanes arepa':
8.00, 'jamon arepa': 7.50
}
```

Save this to a variable called `arepas_menu`.

23.

Next let's create our first *Take a' Arepa* franchise! Our new restaurant is located at `"189 Fitzgerald Avenue"`. Save the `Franchise` object to a variable called `arepas_place`.

Hint

Remember to pass a list consisting of `arepas_menu` from the last task.

24.

Now let's make our new `Business`! The business is called `"Take a' Arepa"`!

25.

Congrats! You created a system of classes that help structure your code and perform all business requirements you need. Whenever we need a new feature we'll have the well-organized code required to make developing and shipping it easy.

26.

If you are stuck on the project or would like to see an experienced developer work through the project, watch the following project walkthrough video!