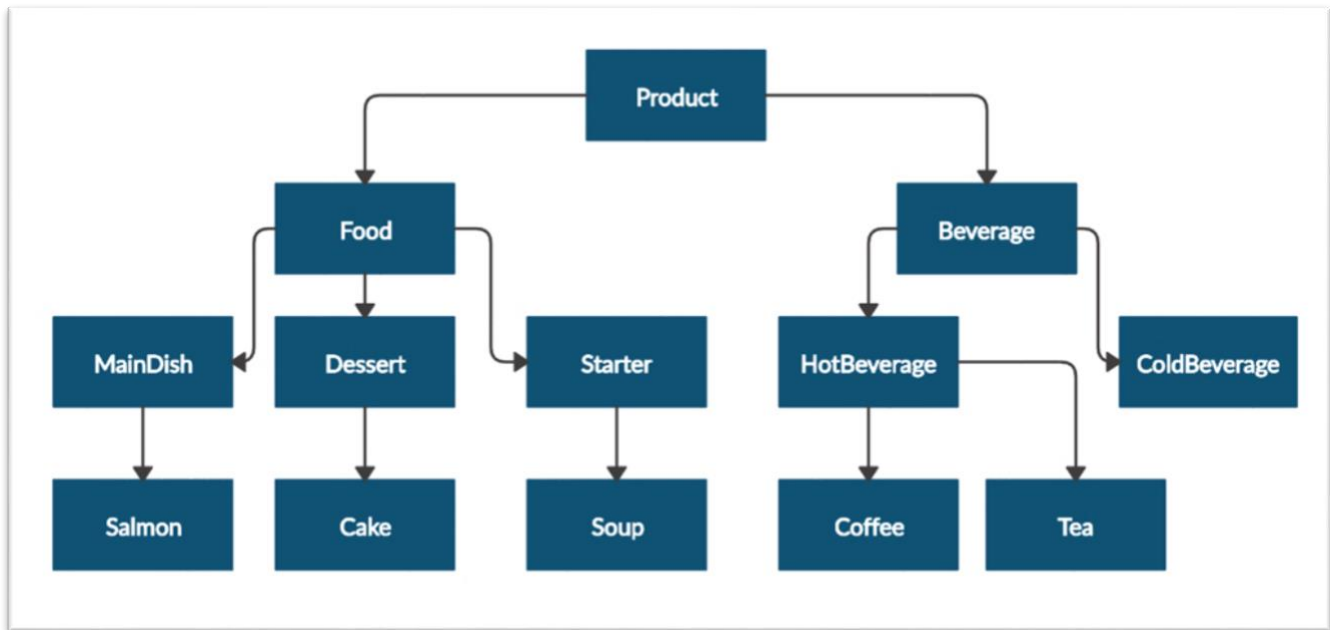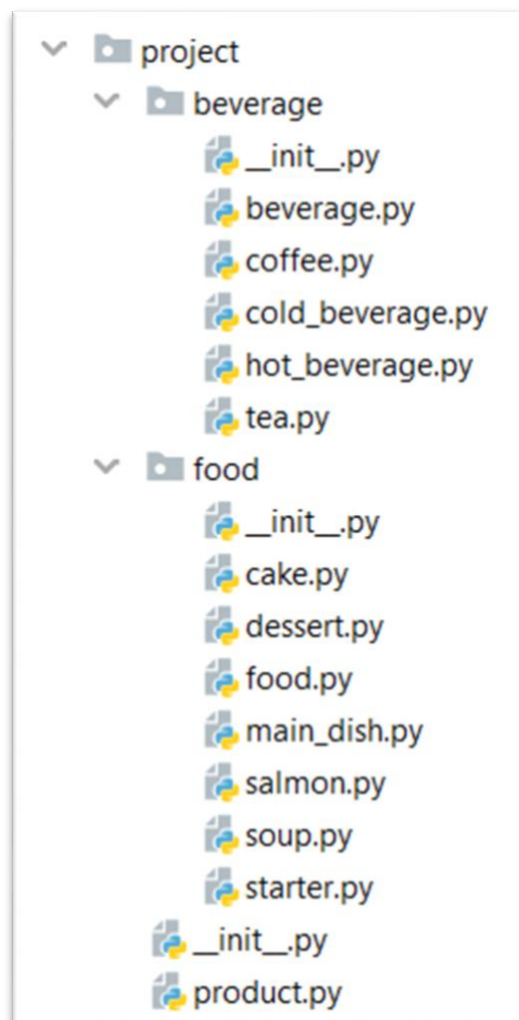Create a **restaurant** with the following classes and hierarchy:

and submit in judge a **zip file**, containing a separate file for each of the classes using the structure shown below:

The **Product** class should have the following **attributes** and subsequent **getters**:

- **name - string**
- **price - float**

Beverage and **Food** classes are **products**.

The **Beverage** class should have the following **attributes** and subsequent **getters**:

- **name - string**
- **price - float**
- **milliliters - float**

The **Food** class should have the following **attributes** and subsequent **getters**:

- **name - string**
- **price - float**
- **grams - float**

HotBeverage and **ColdBeverage** are **beverages** and they accept the **following parameters upon initialization**: **name, price, milliliters**

Coffee and **Tea** are **hot beverages** and they accept the **following parameters upon initialization**: **name, price, milliliters**

The **Coffee** class should have the following **additional attributes** and subsequent **getters**:

- **MILLILITERS = 50 (constant)**
- **PRICE = 3.50 (constant)**
- **caffeine - float**

MainDish, **Dessert** and **Starter** are **food**. They all accept the following parameters upon initialization: **name, price, grams**.

Dessert should accept **one more parameter** in its constructor:

- **calories - float**

Create a **getter** for the attribute **calories**.

Make **Salmon**, **Soup** and **Cake** inherit **MainDish**, **Starter** and **Dessert** classes **respectively**.

A **Cake** must have the following attributes upon initialization:

- **GRAMS = 250 (constant)**
- **CALORIES = 1000 (constant)**
- **PRICE = 5 (constant)**

A **Salmon** should have the following attributes upon initialization:

- **GRAMS = 22 (constant)**

Note: All of the attributes need to be private!

Test Code	Output
product = Product("coffee", 2.5) print(product.__class__.__name__) print(product.name) print(product.price)	Product coffee 2.5 Beverage

beverage = Beverage("coffee", 2.5, 50) print(beverage.__class__.__name__) print(beverage.__class__.__bases__[0].__name__) print(beverage.name) print(beverage.price) print(beverage.milliliters) soup = Soup("fish soup", 9.90, 230) print(soup.__class__.__name__) print(soup.__class__.__bases__[0].__name__) print(soup.name) print(soup.price) print(soup.grams)	Product coffee 2.5 50 Soup Starter fish soup 9.90 230