823G5 Programming in Python

Aim

To familiarise yourself with

- Object Oriented Programming
- Super and sub classes

Exercises

Download the `Lab5Exercises` ZIP file. It contains some basic code for you to develop. The code is a set of Python source code file (`.py`). You will need to open a new PyCharm project and import the `.py` files into it. Refer to the Lab 1 instructions if you cannot remember how to do this.

Hints

Some key points for you to remember:

- A class is a design for code that represents some useful entity that forms a part of a solution to a bigger problem.
- Objects (instances) are instantiated using the notation `x = SomeClass()`. Remember to include the parentheses and any arguments/parameters required.
- A class can be built by sub-classing a parent (base) class. The sub class inherits attributes and methods from its parent class.
- A parent class represents a more generic entity. A sub class represents a more concrete entity.
- Efficient use of sub classes and good OO design reduces code duplication and makes the code a better model of some real-world problem.
- Use of the Python `isinstance()` function is often (but not always) an indication that the OO design can be improved.
- Sub classes can override inherited methods to provide versions that are appropriate in their own specific classes.

You can control the way that Python display numbers in a quite precise way – it will be useful in this lab exercise. The format uses a formatting expression `f`. Here is an example of how to use it…

```
x = 20.2

# Display using 2 decimal places...
print(f'{x:.2f}')

  Displays: 20.20

# Using several variables...

y = 14
print(f'The value of y is {y} and x is {x:.2f}')

  Displays: The value of y is 14 and x is 20.20
```

So, you use the `f` formatter and open a quote mark as usual – this is called an "f-string". You can then refer to a variable (or perform a computation with them) by enclosing the name in curly braces "`{ }`". You can specify formatting instructions by adding ":" and a format code. "`.2f`" means a 'a floating-point number with 2 decimal places' – ideal for currency. Also, this way you don't need to keep on opening and closing quote marks.

Coding challenge

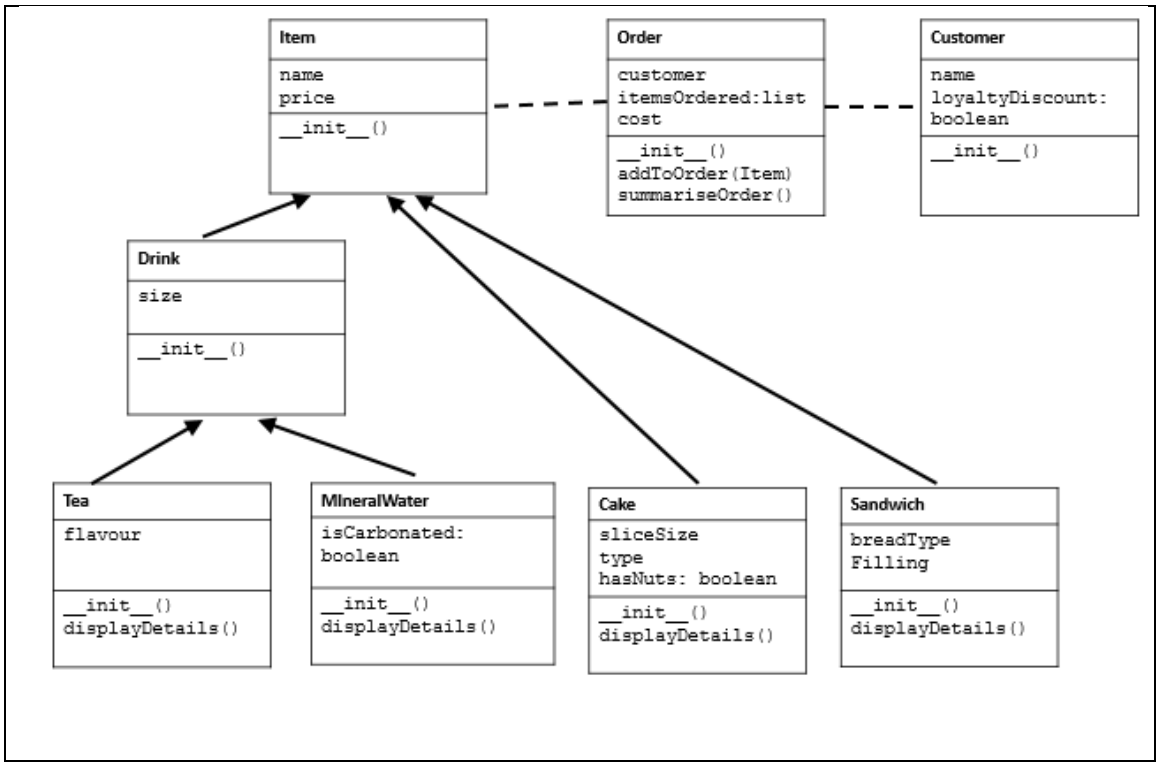
For this challenge, we shall create an OO solution for a small café. The following classes have been deemed relevant to the solution:

- `Customer`: a person that can place an order in the cafe.
- `Order`: a schedule of items ordered and the cost for a particular customer.
- `Item`: a super class for all food and drink that can be ordered in the cafe.
- `Drink`: a sub class of `Item` and serves as a super class for all drink items we can purchase.
- `Tea`: a sub class of `Drink` to represent a cup of tea that we can order.
- `MineralWater`: a sub class of `Drink` to represent bottled water that we can order.
- `Cake`: a sub class of `Item` to represent a tasty item we can order.
- `Sandwich`: a sub class of `Item` to represent a tasty item we can order.

You should also be aware of the following business logic that underpins how the case business operates:

- There is an option for a customer loyalty discount. That data is stored in the `Customer` class.
- Loyal customers get a discount of £0.10 (10p) on all drinks, and £0.20 (20p) on all sandwiches. There is no discount available on cakes, as they are marginal profit items.
- The discount is applied when the order is summarised, so that a loyal customer can see how much money they have saved on their order.
- We need to warn customer clearly if a cake item contains nuts.

Open the `challenge.py` file. It contains the `main()` method, but no class definitions yet. We can see what the intended structure of this solution will be using a class diagram. A class diagram shows the associations between classes and the key attributes and methods:



Your task is to implement the classes with their attributes and methods set out in the class diagram such that the `main()` function can do its job. When you have it working, the output should look something like this:

```
Customer: Harry Palmer
Total items ordered: 2
Black tea £2.00, large, Earl Gray
Club special £4.50, brown, chicken filling
Total cost: £6.50

Customer: Bill Preston
Total items ordered: 3
Evian £1.50, small
Still water
Simple sandwich £1.50, white, cheese filling
Chocolate dream £2.30, medium, chocolate
Warning: contains nuts!
Today you saved £0.30
Total cost: £5.00
```

An example solution is available on Canvas.

Dr. Benjamin Evans
B.D.Evans@sussex.ac.uk