Read all instructions carefully. Not following instructions will result in you not earning the credit you want for the assignment.

Objectives

In Part 5 you will learn the following:

- Override `__str__` magic method to customize the printable version of an object.
- Override `__repr__` magic method to return an unambigous string version of an object that can be executed or evaluated. Often also used for debugging.

Structure

- Module name: desserts
    - Classes
    - Order
    - Candy
    - Cookie
    - IceCream
    - Sundae
- Module name: dessertshop
    - Functions
        - main
    - Classes
        - DessertShop

Problem

In this part of the project, you will override the default magic `__str__` method in all classes to enable printing a receipt using the built-in `print` function. You have implemented `__str__` method in a previous project, but now we present it in the context of overriding the default implementation.

Changes to dessertshop module

- Comment out any lines you may have that print individual Items to the console.
- Replace that code with a single line that uses print to print an order, for example `print(order)`.

Your code must still produce a correct `receipt.pdf` file as in previous parts. If you've done it correctly no changes to it should be needed for Part 5.

Changes to the Concrete Subclasses of DessertItem

Add `__str__` and `__repr__` methods to Candy, Cookie, IceCream, Sundae classes.

Do **not** print in this method. Per the Python docs, this method returns a string to the caller. Using candy corn as an example, formatted Item output would look like this:

```
Candy Corn, 1.5lbs, $0.25/lb, $0.38, 0.03
```

The line gives item name, quantity, price per unit, cost, and tax. Ice Cream Sundaes are different in that they have another line of comma separated values. As always, I suggest using f-strings to build strings like this one.

We format the Item strings this way because it makes the string form easier to process programmatically, while still being human-readable.

Changes to Order Class

- add `__str__` and`__repr__` methods to Order class.

When you print an order to the terminal, it does not need to be in nice columns, just print comma-separated values for each line.

Test Cases

There are no new automated test cases to add. Run your existing automated test cases to make sure they still pass. Then verify by manual inspection that this new version with overridden string methods is correct.

Key Program Requirements

1. No printing to the console occurs in any class other than DessertShop or any function other than main().

2. Each `__str__` method returns a formatted string representation of the corresponding item or order.

3. Submit the following 7 files in your workspace:

- desserts.py
- dessertshop.py
- test_dessert.py
- test_candy.py
- test_cookie.py
- test_icecream.py
- test_sundae.py

Sample Run

1: Candy

2: Cookie

3: Ice Cream

4: Sunday

What would you like to add to the order? (1-4, Enter for done): 2

Enter the type of cookie: Chocolate Chip Macadamia

Enter the quantity purchased: 5

Enter the price per dozen: 4.99

1: Candy

2: Cookie

3: Ice Cream

4: Sunday

What would you like to add to the order? (1-4, Enter for done): 3

Enter the type of ice cream: Mint Chocolate Chip

Enter the number of scoops: 4

Enter the price per scoop: .89

1: Candy

2: Cookie

3: Ice Cream

4: Sunday

What would you like to add to the order? (1-4, Enter for done):4

Enter the type of ice cream: Vanilla

Enter the number of scoops: 3

Enter the price per scoop: .89

Enter the kind of topping used: Sprinkles

Enter the price for the topping: .5

1: Candy

2: Cookie

3: Ice Cream 4: Sunday What would you like to add to the order? (1-4, Enter for done):

Receipt

We show the receipt here as a table so that you know what the PDF receipt should look like.

Name	Quantity	Unit Price	Cost	Tax
Chocolate Chip Macadamia cookies	5 cookies	$4.99/dozen	$2.08	$0.15
Sprinkles Mint Chocolate Chip Sundae	4 scoops	$0.89/scoop	$4.35	$0.32
Sprinkles topping	1	$0.79		
Candy Corn	1.5lbs	$0.25/lb	$0.38	$0.03
Gummy Bears	0.25lbs	$0.35/lb	$0.09	$0.01
Chocolate Chip Cookies	6 cookies	$3.99/dozen	$2.00	$0.14
Pistachio Ice Cream	2 scoops	$0.79/scoop	$1.58	$0.11
Hot Fudge Vanilla Sundae	3 scoops	$0.69/scoop	$3.36	$0.24
Hot Fudge topping	1	$1.29		

Total items in the order	7			
Order Subtotal			$13.84	$1.00
Order Total				$14.84

Correctness

From your terminal, run `ruff check` on each of the 7 files above, such as:

```
ruff check dessert.py
ruff check dessertshop.py
ruff check test_dessert.py
...
```

This will check for syntax errors, violations and many issues that could lead to bugs in your code. Code will be maually graded, so any score received are partial.

Style

From your terminal, run `ruff format` on each of the 7 files above to check the format of your code, like:

```
ruff format dessert.py
ruff format dessertshop.py
ruff format test_dessert.py
```

How to Submit

From your Github assignment repository page, click Submit and enter a nontrivial commit message.

Grading

This project is manually graded. Use the following rubric for grading.

Criteria	Mastery (100)	Developing (75)	Beginning (50)
Overriding of the `__str__` method	The `__str__` method is correctly overridden in all required classes (Candy, Cookie, IceCream, Sundae, Order).	The `__str__` method is overridden in most required classes, but some may be missing or incorrect.	Few or none of the required classes have the `__str__` method correctly overridden.
Formatting of the returned string in the `__str__` method	All overridden `__str__` methods return a string formatted according to the specifications for each class.	Most overridden `__str__` methods return a string formatted correctly, but there may be some discrepancies.	Few or none of the overridden `__str__` methods return a string formatted correctly.
Maintaining functionality of the receipt.pdf file	The receipt.pdf file is generated correctly with no changes needed.	The receipt.pdf file is mostly correct, but some minor modifications may be needed.	The receipt.pdf file is incorrect or not generated at all.
No printing to the console outside of DessertShop and main()	No console printing occurs outside of the `DessertShop` class or `main()` function.	Most console printing is limited to the `DessertShop` class and `main()`, but there may be some instances of printing in other locations.	Console printing occurs outside of the `DessertShop` class or `main()` function.
Running existing test cases	All existing test cases run and pass with no modifications required.	Most existing test cases run and pass without modifications, but some may fail or require changes.	Few or none of the existing test cases run and pass without modifications.

Students should aim for the Mastery level in all categories to ensure they have met all the requirements of the problem. Lower levels represent areas where additional learning and practice may be necessary. As with prior Parts, the overall score is the average of the individual criteria scores.