Exercises Week 3-1 (OOP- classes)

EX.6-1 Class “Cat”

Open VSC then -> File -> Open folder -> select “Python course” folder.

Create two files named "Cat.py" and "main-2.py". The file "main-2.py" will serve as the starting point for your program.

TASK: Imagine that you are embarking on the creation of a computer game. In this exercise, your goal is to create a class named Cat and create multiple instances of this class. Then, thoroughly test all the methods you implement.

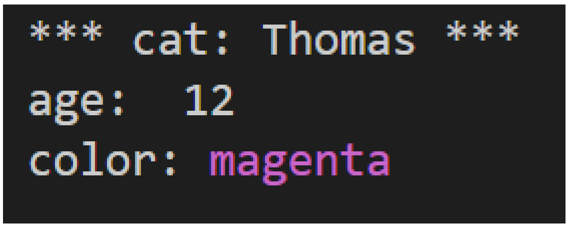
Requirements:

(1) Define the Cat class with the following attributes: **name**, **age**, and **color** (as well as any other attributes you choose to add). Additionally, create the following methods within the Cat class:
display_data(): This method should display the cat's name, age, and color.
display(): This method should draw a cat using ASCII art.
sound(): This method should simulate the sound that the cat makes. It can be just a text message “Meow!” **OR** a sound (use function **Beep**() from **winsound** module)

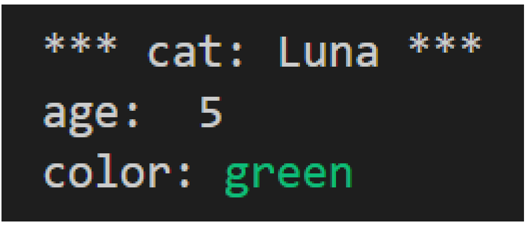
(2) The colour attribute must have one of four values: "blue", "magenta", "green", and "default" (or any other colours you prefer).

(3) Implement a constructor within the Cat class to initialize the attributes when an instance of the class is created.

(4) Examples of methods’ execution:
display_data() – display all attributes:

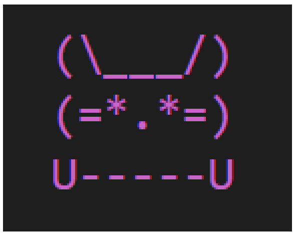


```
*** cat: Thomas ***
age:  12
color: magenta
```

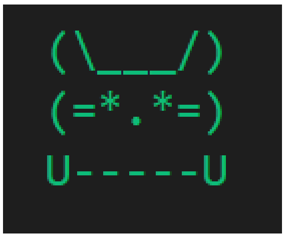


```
*** cat: Luna ***
age:  5
color: green
```

display() – ASCII drawing of the cat with selected colour:

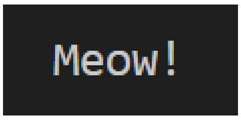


```
(\___/)
(=*.*=)
U-----U
```



```
(\___/)
(=*.*=)
U-----U
```

sound() - display text “Meow!” **OR** play a sound (use function **Beep**() from **winsound** module):



```
Meow!
```

(5) Class Cat must be in a separate module “Cat.py”.

EX.6-2 Class “Product”

Create two files named "Product.py" and "main-3.py". The file "main-3.py" will serve as the starting point for your program.

We are in the process of developing a program to manage stock in a food store. Your task is to design a class named Product that fulfils the specified requirements.

TASK: Design the Product class in a manner that satisfies the following requirements:

Requirements:

(1) Each product should have attributes such as **name**, **price**, **quantity**, and any other attributes you find relevant. Ensure that the Product class includes a constructor that initializes all these attributes.

(2) The Product class must offer the following functionalities:
display(): This function should present all attributes of the product, including its cost, which can be calculated based on the quantity and price.

get_price(): Return the current value of the price attribute.

set_price(new_price): Update the price attribute with the new value provided.

get_quantity(): Return the value of the quantity attribute.

set_quantity(new_quantity): Update the quantity attribute with the new value provided.

sell(quantity_to_sell): Subtract the specified *quantity_to_sell* from the product's quantity attribute. Ensure that the subtraction is performed only if *quantity_to_sell* is less than or equal to the current quantity.

cost(): Return the calculated cost of the product, which is the product of its quantity and price.

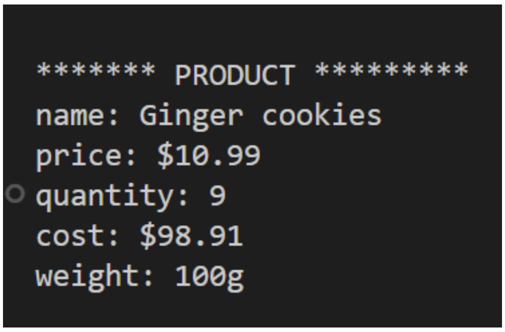
Instructions:

Create the Product class, incorporating the specified attributes and methods as outlined above.

Instantiate several objects of the Product class to represent various products within the food store.

Test each of the functions within the Product instances you've created to confirm that they operate as expected and meet the requirements.

EXAMPLE: display()



```
******* PRODUCT *********
name: Ginger cookies
price: $10.99
quantity: 9
cost: $98.91
weight: 100g
```