

Python Exercise 7

Create a Pet Class and object (30pts)

Write a class called Pet that has the following data and methods:

The data (attributes) for the Pet class is as follows:

- name of the pet
- type of animal (e.g. dog, cat, ferret)
- age of pet

Define your data so that they are hidden from code outside of the class definition.

The Pet class should have the following methods (operations):

- an `__init__` method to set up an object. This method only needs the required parameter.
 - In this method you can initialize the members using any default (hardcoded) values of your choice.
- mutator (setter method) that assigns a name
- mutator (setter method) that assigns the animal type
- mutator (setter method) that assigns the age
- accessor (getter method) that returns the name
- accessor (getter method) that returns the animal type
- accessor (getter method) that returns the age

Next write a main function that creates an object of type Pet. Prompt the user to enter the name, type and age of a pet and use this data to update the attributes of the Pet object. Use the mutator methods to pass along the information of the pet, and the accessor methods to confirm that all of the data was properly stored. The main function may be located in the same file as the class definition.

Sample output is shown below.

A pet object has been created. Here is the initial information about the pet:

Name of pet: Not provided

Type of pet: Not provided

Age of pet: 0

Let's update the information for a pet!

Enter the pet's name: Pico

Enter the type of animal: dog

Enter the pet's age: 6

Here is the updated information about the pet:

Name of pet: Pico

Type of pet: dog

Age of pet: 6

Tips:

- Your output does not have to match the text exactly, but your code should use all methods created from your class. In the sample output, accessors are used to display initial and updated information about the pet, the mutators are used (with local variables) to update the information, and the initializer method is used automatically when the object is initially created.
- Feel free to create more than one object for the pets in your life! However, you must create at least one object to test all your methods.

Retail items (30pts)

Write a class called `Retail_Item` that has the following data and methods:

The data (attributes) for the `Retail_Item` Class is as follows:

- type of item (e.g. Jacket, shirt)
- amount of item in inventory
- price of item

Define your data so that they are hidden from code outside of the class definition.

Create two methods for this class:

- an `__init__` method that accepts three parameters (corresponding to the data) in addition to the required parameter.
- a `__str__` method that formats the data (in preparation for output).
 - Create formatting strings with field widths and justifications in this part.

Next write a main function that does the following:

- Prompts the user to enter the information for two items
 - You are welcome to create more, but two items are the minimum
- Creates an object to represent each item
 - Note: No loop or list is required for this question
- Displays the information of the two items in a neat table
 - The table header should be created in main so that it aligns with the formatting for the object's data. (Remember, the `__str__` method should be used here.)

The main function may be located in the same file as the class definition.

Sample output is shown below:

Name of item 1: Jeans

Amount of item 1: 50

Price of item 1: 59.95

Name of item 2: T-Shirt

Amount of item 2: 65

Price of item 2: 19.99

Here is a summary of the items you added:

Item	Amount	Price
Jeans	50	\$59.95
T-Shirt	65	\$19.99