Foundations of Computer Science A

Week 9

Exercise 1

Design a class called CarRental that conforms to the following requirements:

1. The class should represent a car with three private attributes: ‘brand’, ‘model’, and ‘daily_rate’.
2. An ‘__init__’ method for the class with parameters to initialize the ‘brand’, ‘model’, and ‘daily_rate’ attributes.
3. A magic/dunder method that prints the car's brand and model when ‘print(car)’ is called.
4. A magic/dunder method "greater than" to compare two cars' daily rental rates.
5. A magic/dunder method "less than" to compare two cars' daily rental rates.
6. A magic/dunder method "equal to" to compare two cars' daily rental rates.
7. A method ‘rent_for_days’ that takes the number of days as a parameter and returns the total rental cost.

For example:

```
car1 = CarRental(‘Toyota’, ’Corolla’, daily_rate = 100)
car2 = CarRental(‘Ford’, ’Mustang’, daily_rate = 150)

print(car1)        # Toyota Corolla
print(car1 > car2)    # False
print(car1 < car2)    # True
print(car1 == car2)    # False
print(car1.rent_for_days(5)) # 500
```

Exercise 2

Create a base class called ‘Solid’. This class should have three empty methods: ‘volume’, ‘surface_area’, and ‘__str__’. The ‘__str__’ method should return the name of the class.

Now create two more classes, ‘Sphere’ and ‘Cuboid’, with the following specifications:

‘Sphere’:

1. Inherits from the ‘Solid’ class.
2. Has an attribute called ‘radius’ initialized through the constructor.
3. Overrides the ‘volume’ method to return the volume of a sphere (rounded to 2 decimal places):

$$\frac{4}{3}\pi \times radius^3$$

4. Overrides the ‘surface_area’ method to return the surface area of a sphere (rounded to 2 decimal places):

$$4\pi \times radius^2$$

‘Cuboid’:

1. Inherits from ‘Solid’ class.
2. Has attributes ‘length’, ‘width’, and ‘height’ initialized through the constructor.
3. Overrides the ‘volume’ method to return the volume of a cuboid: length×width×height
4. Overrides the ‘surface_area’ method to return the surface area of a cuboid: 2(length×width+width×height+height×length).

Instantiate a ‘Sphere’ object and a ‘Cuboid’ object, and add them to a list called ‘solids’. Iterate through this list, printing the volume and surface area of each solid in the format:

"name of class volume is volume of object"
"name of class surface area is surface area of object"

from math import pi

Exercise 3

You are responsible for designing a software system for an electronics vending machine. The machine dispenses gadgets like USB drives, earphones, and portable chargers. Each item has a different price and unique attributes. Design an object-oriented solution for this scenario with the following classes:

1. Gadget:

✧ ‘name’ (string): the name of the gadget.
✧ ‘price’ (float): the price of the gadget.
✧ A constructor to set the attributes ‘name’ and ‘price’.

2. USBDrive:

✧ Inherits from the Gadget class.
✧ ‘storage’ (integer): the storage capacity in GB.
✧ A constructor that initializes these values from arguments. Demonstrate how to use the parent class to initialize ‘name’ and ‘price’.

3. Earphones:

✧ Inherits from the Gadget class.
✧ ‘wireless’ (boolean): indicates if the earphones are wireless.
✧ A constructor that initializes these values.

4. PortableCharger:

✧ Inherits from the Gadget class.
✧ ‘capacity’ (float): the battery capacity in mAh.
✧ A constructor that initializes these values.

5. ElectronicsVendingMachine:

✧ ‘inventory’ (list): a list of Gadget objects initialized to an empty list.
✧ ‘id’ (string): unique identifier for the vending machine.

- A method ‘add’ to add instances of Gadget to the vending machine inventory list.
- A method ‘dispense‘ to search the inventory by the name of a gadget and remove the gadget from the inventory.
- A ‘__str__’ method to print the vending machine id and the current contents in the inventory.