

Object Oriented Programming

The following exercises come from *Starting Out With Python 2E*, by Tony Gaddis, Pearson Publishing.

1. Pet Class

Write a class named `Pet`, which should have the following data attributes:

- `__name` (for the name of a pet)
- `__animal_type` (for the type of animal that a pet is. Example values are 'Dog', 'Cat', and 'Bird')
- `__age` (for the pet's age)

The `Pet` class should have an `__init__` method that creates these attributes. It should also have the following methods:

- `set_name`
This method assigns a value to the `__name` field.
- `set_animal_type`
This method assigns a value to the `__animal_type` field.
- `set_age`
This method assigns a value to the `__age` field.
- `get_name`
This method returns the value of the name field.
- `get_type`
This method returns the value of the type field.
- `get_age`
This method returns the value of the age field.

Once you have written the class, write a program that creates an object of the class and prompts the user to enter the name, type, and age of his or her pet. This data should be stored as the object's attributes. Use the object's accessor methods to retrieve the pet's name, type, and age and display this data on the screen.

2. Car Class

Write a class named Car that has the following data attributes:

- `__year_model` (for the car's year model)
- `__make` (for the make of the car)
- `__speed` (for the car's current speed)

The Car class should have an `__init__` method that accepts the car's year model and make as arguments. These values should be assigned to the object's `__year_model` and `__make` data attributes. It should also assign 0 to the `__speed` data attribute.

The class should also have the following methods:

- `accelerate`

The `accelerate` method should add 5 to the speed data attribute each time it is called.

- `brake`

The `brake` method should subtract 5 from the speed data attribute each time it is called.

- `get_speed`

The `get_speed` method should return the current speed.

Next, design a program that creates a Car object, and then calls the `accelerate` method five times. After each call to the `accelerate` method, get the current speed of the car and display it. Then call the `brake` method five times. After each call to the `brake` method, get the current speed of the car and display it.

3. Employee Management System

Write a class named `Employee` that holds the following data about an employee in attributes: name, ID number, department, and job title. Once you have written the class, write a program that creates three `Employee` objects to hold the following data:

Name	ID Number	Department	Job Title
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

The program should store this data in a dictionary. Use the employee ID number as the key. The program should present a menu that lets the user perform the following actions:

- Look up an employee in the dictionary
- Add a new employee to the dictionary
- Change an existing employee's name, department, and job title in the dictionary
- Delete an employee from the dictionary
- Quit the program

When the program ends, it should pickle the dictionary and save it to a file. Each time the program starts, it should try to load the pickled dictionary from the file. If the file does not exist, the program should start with an empty dictionary.

4. Trivia Game

In this exercise you will create a simple trivia game for two players. The program will work as follows.

Starting with player 1, each player gets a turn at answering 5 trivia questions. (There should be a total of 10 questions.) When a question is displayed, 4 possible answers are also displayed. Only one of the answers is correct, and if the player selects the correct answer, he or she earns a point. After answers have been selected for all the questions, the program displays the number of points earned by each player and declares the player with the highest number of points the winner.

To create this program, write a Question class to hold the data for a trivia question. The Question class should have attributes for the following data:

- A trivia question
- Possible answer 1
- Possible answer 2
- Possible answer 3
- Possible answer 4
- The number of the correct answer (1, 2, 3, or 4)

The Question class also should have an appropriate `__init__` method, accessors, and mutators.

The program should have a list or a dictionary containing 10 Question objects, one for each trivia question. Make up your own trivia questions on the subject or subjects of your choice for the objects.

5. Employee and ProductionWorker Classes

Write an `Employee` class that keeps data attributes for the following pieces of information:

- Employee name
- Employee number

Next, write a class named `ProductionWorker` that is a subclass of the `Employee` class. The `ProductionWorker` class should keep data attributes for the following information:

- Shift number (an integer, such as 1, 2, or 3)
- Hourly pay rate

The workday is divided into two shifts: day and night. The shift attribute will hold an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Write the appropriate accessor and mutator methods for each class.

Once you have written the classes, write a program that creates an object of the `ProductionWorker` class and prompts the user to enter data for each of the object's data attributes. Store the data in the object and then use the object's accessor methods to retrieve it and display it on the screen.

6. ShiftSupervisor Class

In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. Write a `ShiftSupervisor` class that is a subclass of the `Employee` class you created in Programming Exercise 5. The `ShiftSupervisor` class should keep a data attribute for the annual salary and a data attribute for the annual production bonus that a shift supervisor has earned. Demonstrate the class by writing a program that uses a `ShiftSupervisor` object.