

Week 6

Introduction to OOP, classes and objects

Part 1. Theoretical questions

1.1 Multiple choice questions

1. The _____ programming practice is centered on creating functions that are separate from the data that they work on.
 - a. modular
 - b. procedural
 - c. functional
 - d. object-oriented
2. The _____ programming practice is centered on creating objects.
 - a. object-centric
 - b. objective
 - c. procedural
 - d. object-oriented
3. A(n) _____ is a component of a class that references data.
 - a. method
 - b. instance
 - c. data attribute
 - d. module
4. An object is a(n) _____.
 - a. blueprint
 - b. cookie cutter
 - c. variable
 - d. instance
5. By doing this you can hide a class's attribute from code outside the class.
 - a. avoid using the self parameter to create the attribute
 - b. begin the attribute's name with two underscores
 - c. begin the name of the attribute with private__
 - d. begin the name of the attribute with the @ symbol
6. A(n) _____ method gets the value of a data attribute but does not change it.
 - a. retriever
 - b. constructor
 - c. mutator
 - d. accessor
7. A(n) _____ method stores a value in a data attribute or changes its value in some other way.
 - a. modifier
 - b. constructor
 - c. mutator
 - d. accessor
8. The _____ method is automatically called when an object is created.
 - a. __init__
 - b. init
 - c. __str__
 - d. __object__
9. If a class has a method named __str__, which of these is a way to call the method?
 - a. you call it like any other method: object.__str__()
 - b. by passing an instance of the class to the built in str function
 - c. the method is automatically called when the object is created
 - d. by passing an instance of the class to the built-in state function
10. A set of standard diagrams for graphically depicting object-oriented systems is provided by _____.

- a. the Unified Modeling Language
- b. flowcharts
- c. pseudocode
- d. the Object Hierarchy System

11. In one approach to identifying the classes in a problem, the programmer identifies the _____ in a description of the problem domain.

- a. verbs
- b. adjectives
- c. adverbs
- d. nouns

12. In one approach to identifying a class's data attributes and methods, the programmer identifies the class's _____.

- a. responsibilities
- b. name
- c. synonyms
- d. nouns

1.2 True or False

1. The practice of procedural programming is centered on the creation of objects.
2. Object reusability has been a factor in the increased use of object-oriented programming.
3. It is a common practice in object-oriented programming to make all of a class's data attributes accessible to statements outside the class.
4. A class method does not have to have a self-parameter.
5. Starting an attribute name with two underscores will hide the attribute from code outside the class.
6. You cannot directly call the `__str__` method.
7. One way to find the classes needed for an object-oriented program is to identify all of the verbs in a description of the problem domain.

1.3 Short Answer

1. What is encapsulation?
2. Why should an object's data attributes be hidden from code outside the class?
3. What is the difference between a class and an instance of a class?
4. The following statement calls an object's method. What is the name of the method?
What is the name of the variable that references the object? `wallet.get_dollar()`
5. When the `__init__` method executes, what does the `self` parameter reference?
6. In a Python class, how do you hide an attribute from code outside the class?
7. How do you call the `__str__` method?

Part 2 Programming Exercise

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 accept 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. Personal Information Class

Design a class that holds the following personal data: name, address, age, and phone number. Write appropriate accessor and mutator methods. Also, write a program that creates three instances of the class. One instance should hold your information, and the other two should hold your friends' or family members' information.

4. Employee Class

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 the three objects and then display the data for each employee on the screen.

5. RetailItem Class

Write a class named `RetailItem` that holds data about an item in a retail store. The class should store the following data in attributes: item description, units in inventory, and price. Once you have written the class, write a program that creates three `RetailItem` objects and stores the following data in them:

	Description	Units in Inventory	Price
Item #1	Jacket	12	59.95
Item #2	Designer Jeans	40	34.95
Item #3	Shirt	20	24.95

6. Employee Management System

This exercise assumes that you have created the `Employee` class for Programming Exercise 4. Create a program that stores `Employee` objects 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.

7. Cash Register

This exercise assumes that you have created the `RetailItem` class for Programming Exercise 5. Create a `CashRegister` class that can be used with the `RetailItem` class. The `CashRegister` class should be able to internally keep a list of `RetailItem` objects. The class should have the following methods:

- A method named `purchase_item` that accepts a `RetailItem` object as an argument. Each time the `purchase_item` method is called, the `RetailItem` object that is passed as an argument should be added to the list.
- A method named `get_total` that returns the total price of all the `RetailItem` objects stored in the `CashRegister` object's internal list.
- A method named `show_items` that displays data about the `RetailItem` objects stored in the `CashRegister` object's internal list.
- A method named `clear` that should clear the `CashRegister` object's internal list.

Demonstrate the `CashRegister` class in a program that allows the user to select several items for purchase. When the user is ready to check out, the program should display a list of all the items he or she has selected for purchase, as well as the total price.

8. Trivia Game

In this programming exercise you will create a simple trivia game for two players. The program will work like this:

- 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.