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~~

~~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.~~

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?

2. Write a class definition named Book. The Book class should have data attributes for a book's title, the author's name, and the publisher's name. The class should also have the following:
   a. An __init__ method for the class. The method should accept an argument for each of the data attributes.
   b. Accessor and mutator methods for each data attribute.
   c. An __str__ method that returns a string indicating the state of the object.
3. ~~Look at the following description of a problem domain:~~

   ~~The bank offers the following types of accounts to its customers: savings accounts, checking accounts, and money market accounts. Customers are allowed to deposit money into an account (thereby increasing its balance), withdraw money from an account (thereby decreasing its balance), and earn interest on the account. Each account has an interest rate.~~

   ~~Assume that you are writing a program that will calculate the amount of interest earned for a bank account.~~
   a. ~~Identify the potential classes in this problem domain.~~
   b. ~~Refine the list to include only the necessary class or classes for this problem.~~
   c. ~~Identify the responsibilities of the class or classes.~~

Programming Exercises

1. Pet Class

VideoNote
The 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_animal_type
  This method returns the value of the __animal_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 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, then display the data for each employee on the screen.