HW 3 \_Farvardin\_#Mahdi-Karimi#

1. What is the difference between a class and an object?  
  
In object-oriented programming, a class is a blueprint or a template for creating objects, while an object is an instance of a class. A class defines a set of attributes and methods that are common to all objects of that class. It specifies the data and behavior of the objects that will be created from it. On the other hand, an object is a specific instance of a class that has its own unique set of values for the attributes defined in the class. In other words, a class is a general concept, while an object is a specific implementation of that concept. Classes are used to create objects, and objects are used to perform tasks and store data in a program.  
  
2. What are some other names for the term instance variable?  
  
Some other names for the term instance variable include member variable, field, attribute, and non-static variable.  
  
3. What is another name for the term method?  
  
In Python, a method is a function that is associated with an object and is called by its name. The first parameter of a method definition is always "self"1  
. However, the search results do not provide another name for the term "method" in Python.  
  
4. What symbol associates an object with a method invocation?  
  
The symbol that associates an object with a method invocation in Python is the dot "."  
. When a method is called, it is associated with an object and the dot notation is used to specify the object that the method is called on  
. The "@" symbol in Python is mainly used for decorators, which are functions that modify the behavior of other functions  
  
5. How does a method differ from a function?  
  
In Python, a method is a function that is associated with an object and is called by its name. A method is implicitly passed to the object on which it is invoked and always includes "self" as its first parameter. A method can operate on the data (instance variables) that is contained by the corresponding class. On the other hand, a function is not linked to anything and can be executed just by calling it by its name. Functions are defined independently and can be called by their name only. The key difference between a method and a function is the concept of "Class and its Object"  
  
6. What method from the string class returns a new string with no leading or trailing whitespace?  
  
The method from the string class that returns a new string with no leading or trailing whitespace is the trim() method. This method removes both leading and trailing whitespace from a string and returns a copy of the string with the whitespace removed.  
  
7. What function returns the length of its string argument?  
The len() function is used to return the length of its string argument in Python1  
. It is a built-in function in Python and can be used to get the length of a string, list, tuple, or any other iterable object  
  
8. What type of object does the open function return?  
  
The open() function in Python is used to open a file and returns it as a file object.  
    The file object returned by the open() function can be used to read, write, or modify the contents of the file1 .  
  
9. What does the second parameter of the open function represent?  
  
The second parameter of the open() function in Python represents the mode in which the file is opened.  
The mode parameter is an optional parameter and its default value is 'r', which stands for read mode.  
The mode parameter can be set to different values depending on the type of access needed for the file. Here are some of the most commonly used modes:  
  
    'r': read mode (default)  
    'w': write mode  
    'a': append mode  
    'x': exclusive creation mode  
    'b': binary mode  
    't': text mode  
  
The mode parameter can also be set to a combination of these modes. For example, 'rb' would open the file in binary mode for reading, while 'w+' would open the file in write mode and also allow reading.  
  
10. Write a program that stores the first 100 integers to a text file named numbers.txt. Each number   
should appear on a line all by itself.  
  
with open("numbers.txt", "w") as file:  
    for i in range(1, 101):  
        file.write(str(i) + "\n")  
  
Description:  
   This program uses the open() function to open a file named numbers.txt in write mode and assigns the returned file object to a variable named file. It then uses a for loop to iterate over the range of integers from 1 to 100 and writes each integer to a new line in the file using the write() method of the file object. The str() function is used to convert the integer to a string before writing it to the file, and the "\n" character is used to add a newline after each number.  
  
11. Complete the following function that reads a collection of integers from a text file named numbers.txt.   
Each number in the file appears on a line all by itself. The function accepts a single parameter, a string   
text file name. The function returns the sum of the integers in the file.   
def sumfile(filename):  
  
Here's a Python function that reads a collection of integers from a text file named numbers.txt and returns the sum of the integers:  
  
def sumfile(filename):  
    with open(filename, "r") as file:  
        numbers = [int(line.strip()) for line in file]  
        return sum(numbers)  
  
This function uses the open() function to open the file specified by the filename parameter in read mode and assigns the returned file object to a variable named file. It then uses a list comprehension to read each line of the file, convert it to an integer using the int() function, and strip any whitespace characters using the strip() method. The resulting list of integers is then passed to the sum() function, which returns the sum of the integers. To use this function, simply call it with the name of the file containing the integers:  
  
total = sumfile("numbers.txt")  
print("The sum of the integers is:", total)  
  
This program calls the sumfile() function with the filename "numbers.txt" and assigns the returned sum to a variable named total. It then prints the sum to the console. Note that if the file specified in the sumfile() function does not exist or contains non-integer values, the program will raise an error.  
  
12. Provide the syntactic sugar for each of the following methods of the Fraction class:   
(a) *sub* : ' - '  
  
f1 = Fraction(1, 2)  
f2 = Fraction(1, 3)  
f3 = f1 - f2  
  
(b) *eq* : ' == '  
  
f1 = Fraction(1, 2)  
f2 = Fraction(2, 4)  
if f1 == f2:  
    print("The fractions are equal")  
else:  
    print("The fractions are not equal")  
  
(c) *neg* : ' -f '  
  
f1 = Fraction(1, 2)  
f2 = -f1  
  
(d) *gt* : ' > '  
  
f1 = Fraction(1, 2)  
f2 = Fraction(1, 3)  
if f1 > f2:  
    print("f1 is greater than f2")  
else:  
    print("f2 is greater than or equal to f1")  
  
13. How is using a Turtle object from Python’s Turtle graphics module different from using the free   
functions; for example, t.penup() versus penup()?  
  
The Turtle graphics module in Python provides both object-oriented and procedure-oriented ways to create turtle graphics primitives.  
  
import turtle  
  
t = turtle.Turtle()  
t.forward(100)  
  
Instead, we can call functions directly to perform various actions. For example, to move the turtle using the penup() function, do the following:  
  
import turtle  
  
turtle.penup()  
turtle.forward(100)  
  
14. For each of the drawings below write a program that draws the shape using a Turtle object from   
Python’s Turtle graphics module.  
  
  
    Circle:  
  
import turtle  
  
t = turtle.Turtle()  
t.circle(50)  
  
  
    Equilateral triangle:  
  
import turtle  
  
t = turtle.Turtle()  
for i in range(3):  
    t.forward(100)  
    t.left(120)  
  
Table 5×5 :  
  
import turtle  
  
t = turtle.Turtle()  
for i in range(5):  
    for j in range(5):  
        t.penup()  
        t.goto(j \* 50, i \* 50)  
        t.pendown()  
        t.forward(50)  
        t.right(90)  
        t.forward(50)  
        t.right(90)  
        t.forward(50)  
        t.right(90)  
        t.forward(50)  
  
Horizontal sawtooth:  
  
import turtle  
  
t = turtle.Turtle()  
for i in range(5):  
    t.forward(50)  
    t.right(90)  
    t.forward(50)  
    t.left(90)  
  
five-pointed star:  
  
import turtle  
  
t = turtle.Turtle()  
for i in range(5):  
    t.forward(100)  
    t.right(144)  
  
15. Does Python permit a programmer to change one symbol in a string object? If so, how?  
  
No, Python does not permit a programmer to change one symbol in a string object. Strings in Python are immutable, which means that once a string is created, its contents cannot be changed. However, it is possible to create a new string that is a modified version of the original string.  
  
16. What would be the consequences if a turtle.Turtle object were immutable  
  
If a turtle.Turtle object were immutable, it would not be possible to change its state after it is created. This would mean that once a turtle is created, its position, orientation, and other properties could not be modified. This would severely limit the usefulness of the turtle module for drawing complex graphics, as it would not be possible to move the turtle to different positions on the canvas, change its direction, or change its appearance.  
  
17. In the context of programming, what is garbage?  
  
In the context of programming, "garbage" refers to memory that has been allocated to a program but is no longer being used or referenced by the program. This can occur when a program creates objects or variables in memory, but then fails to release them when they are no longer needed. Over time, this can lead to a buildup of unused memory, which can slow down the program and eventually cause it to crash.  
  
18. What is garbage collection, and how does it work in Python?  
  
Garbage collection is the process of automatically freeing up memory that is no longer being used by a program. In Python, garbage collection works automatically and is performed by the Python interpreter. When an object is created in Python, the interpreter keeps track of how many references there are to that object. When there are no more references to an object, the garbage collector frees up the memory used by that object  
  
19. Consider the following code:   
a = "ABC"   
b = a   
c = b   
a = "XYZ"  
  
(a) At the end of this code’s execution what is the reference count for the string object "ABC"?  
    At the end of the execution of the given code, the reference count for the string object "ABC" is 1.  
  
(b) At the end of this code’s execution is b an alias of a?  
   No, 'b' is not an alias of 'a' at the end of the execution of the given code.  
  
(c) At the end of this code’s execution is b an alias of c?  
   No, 'b' is not an alias of 'c' at the end of the execution of the given code.