1. Make a class called Thing with no contents and print it. Then, create an object called example

from this class and also print it. Are the printed values the same or different?

ANS In Python, everything is an object. So, when you use the type() function to print the type of the value stored in a variable to the console, it returns the class type of the object. For instance, if the type is a string and you use the type() on it, you'd get <class 'string'> as the result.

2. Create a new class called Thing2 and add the value &#39;abc&#39; to the letters class attribute. Letters

should be printed.

ANS Adding attributes to a Python class is very straight forward, you just use the '. ' operator after an instance of the class with whatever arbitrary name you want the attribute to be called, followed by its value.

3. Make yet another class called, of course, Thing3. This time, assign the value &#39;xyz&#39; to an instance

(object) attribute called letters. Print letters. Do you need to make an object from the class to do

this?

ANS A method is termed as a function that belongs to an object or class in python language. A method is defined as a procedure that is derived from the basic oops concept that is object-oriented programing. Whereas a function is a set of c.o.d.e.s that are reusable, which can be called or accessed anywhere in a program.

4. Create an Element class with the instance attributes name, symbol, and number. Create a class

object with the values &#39;Hydrogen,&#39; &#39;H,&#39; and 1.

ANS Instance Attribute. Let's start with the basics: An instance attribute is a Python variable belonging to one, and only one, object. This variable is only accessible in the scope of this object, and it's defined inside the constructor function, \_\_init\_\_(self,..) of the class.

5. Make a dictionary with these keys and values: &#39;name&#39;: &#39;Hydrogen&#39;, &#39;symbol&#39;: &#39;H&#39;, &#39;number&#39;: 1. Then,

create an object called hydrogen from class Element using this dictionary.

ANS n Python, dictionaries are mutable data structures that allow you to store key-value pairs. Dictionary can be created using the dict() constructor or curly braces' {}'. Once you have created a dictionary, you can add, remove, or update elements using the methods dict. update(), dict.27 Feb 2023

6. For the Element class, define a method called dump() that prints the values of the object’s

attributes (name, symbol, and number). Create the hydrogen object from this new definition and

use dump() to print its attributes.

ANS \_init\_\_ : the initialisation method of an object, which is called when the object is created. \_\_str\_\_ : the string representation method of an object, which is called when you use the str function to convert that object to a string.

7. Call print(hydrogen). In the definition of Element, change the name of method dump to \_\_str\_\_,

create a new hydrogen object, and call print(hydrogen) again.

ANS The name derives from the Greek hydro for "water" and genes for "forming" because it burned in air to form water. Hydrogen was discovered by the English physicist Henry Cavendish in 1766. Scientists had been producing hydrogen for years before it was recognized as an element.

8. Modify Element to make the attributes name, symbol, and number private. Define a getter

property for each to return its value.

ANS Getters: These are the methods used in Object-Oriented Programming (OOPS) which helps to access the private attributes from a class. Setters: These are the methods used in OOPS feature which helps to set the value to private attributes in a class.

9. Define three classes: Bear, Rabbit, and Octothorpe. For each, define only one method: eats(). This

should return &#39;berries&#39; (Bear), &#39;clover&#39; (Rabbit), or &#39;campers&#39; (Octothorpe). Create one object from

each and print what it eats.

10. Define these classes: Laser, Claw, and SmartPhone. Each has only one method: does(). This

returns &#39;disintegrate&#39; (Laser), &#39;crush&#39; (Claw), or &#39;ring&#39; (SmartPhone). Then, define the class Robot that

has one instance (object) of each of these. Define a does() method for the Robot that prints what its

component objects do.

ANS Make a class called Thing with no contents and print it. Then, create an object called example from this class and also print it. Are the printed values the same or different?

2. Make a new class called Thing2 and assign the value 'abc' to a class attribute called letters. Print letters.

3. Make yet another class called Thing3. This time, assign the value 'xyz' to an instance (object) attribute called letters. Print letters. Do you need to make an object from the class to do this?

4. Make a class called Element, with instance attributes name, symbol, and number. Create an object of this class with the values 'Hydrogen', 'H', and 1.

5. For the Element class, define a method called dump() that prints the values of the object's attributes (name, symbol, and number). Create the hydrogen object from this new definition and use dump() to print its attributes.

6. Call print(hydrogen). In the definition of Element, change the name of the method dump to \_\_str\_\_, create a new hydrogen object, and call print(hydrogen) again.

7. Modify Element to make the attributes name, symbol, and number private. Define a getter property for each to return its value.

8. Define three classes: Bear, Rabbit, and Octothorpe. For each, define only one method: eats(). This should return 'berries' (Bear), 'clover' (Rabbit), or 'campers' (Octothorpe). Create one object from each and print what it eats.

9. Define these classes: Laser, Claw, and SmartPhone. Each has only one method: does(). This returns 'disintegrate' (Laser), 'crush' (Claw), or 'ring' (SmartPhone). Then, define the class Robot that has one instance (object) of each of these. Define a does() method for the Robot that prints what its component objects do.

FINISH\*\*\*