**Assignment 19**

**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?**

In [7]:

**class** Thing:

**pass**

In [8]:

print(Thing)

<class '\_\_main\_\_.Thing'>

In [6]:

example**=**Thing()

print(example)

<\_\_main\_\_.Thing object at 0x00000207687A6808>

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

In [11]:

**class** Thing2:

**def** \_\_init\_\_(self,letters**=**'abc'):

self**.**letter**=**letters

ob**=**Thing2()

In [12]:

print(ob**.**letter)

abc

**3. Make yet another class called, of course, 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?**

In [13]:

**class** Thing3:

**def** \_\_init\_\_(self,letters):

self**.**letter**=**letters

ob**=**Thing3('xyz')

In [14]:

print(ob**.**letter)

xyz

Yes we 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.**

In [16]:

**class** Element:

**def** \_\_init\_\_(self,name,symbol,number):

self**.**name**=**name

self**.**symbol**=**symbol

self**.**number**=**number

ob**=**Element('Hydrogen','H',1)

**5. Make a dictionary with these keys and values: 'name': 'Hydrogen', 'symbol': 'H', 'number': 1. Then, create an object called hydrogen from class Element using this dictionary.**

In [22]:

d**=**{'name':'Hydrogen','symbol':'H','number':1}

hydrogen**=**Element(d['name'],d['symbol'],d['number'])

**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.**

In [59]:

**class** Element:

**def** \_\_init\_\_(self,name,symbol,number):

self**.**name**=**name

self**.**symbol**=**symbol

self**.**number**=**number

**def** dump(self):

**return** f'{self**.**name,self**.**symbol, self**.**number}'

In [60]:

hydrogen**=**Element(d['name'],d['symbol'],d['number'])

In [62]:

print(hydrogen**.**dump())

('Hydrogen', 'H', 1)

**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.**

In [46]:

print(hydrogen)

<\_\_main\_\_.Element object at 0x000002076988C5C8>

In [56]:

**class** Element:

**def** \_\_init\_\_(self,name,symbol,number):

self**.**name**=**name

self**.**symbol**=**symbol

self**.**number**=**number

**def** \_\_str\_\_(self):

**return** f'{self**.**name, self**.**symbol, self**.**number}'

In [57]:

hydrogen**=**Element(d['name'],d['symbol'],d['number'])

In [58]:

print(hydrogen)

('Hydrogen', 'H', 1)

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

In [69]:

**class** Element:

**def** \_\_init\_\_(self,name,symbol,number):

self**.**\_\_name**=**name

self**.**\_\_symbol**=**symbol

self**.**\_\_number**=**number

**def** dump(self):

**return** f'{self**.**\_\_name,self**.**\_\_symbol, self**.**\_\_number}'

In [70]:

ob**=**Element('Hydrogen','H',1)

In [72]:

ob**.**\_\_dict\_\_

Out[72]:

{'\_Element\_\_name': 'Hydrogen', '\_Element\_\_symbol': 'H', '\_Element\_\_number': 1}

In [74]:

print(ob**.**\_Element\_\_name)

print(ob**.**\_Element\_\_symbol)

print(ob**.**\_Element\_\_number)

Hydrogen

H

1

**9. 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.**

In [79]:

**class** Bear:

**def** eat(self):

**return** 'berries'

**class** Rabbit:

**def** eat(self):

**return** 'clover'

**class** Octothorpe:

**def** eat(self):

**return** 'campers'

In [80]:

ob\_bear**=**Bear()

ob\_Rabbit**=**Rabbit()

ob\_Octo**=**Octothorpe()

In [81]:

ob\_bear**.**eat()

Out[81]:

'berries'

In [82]:

ob\_Rabbit**.**eat()

Out[82]:

'clover'

In [83]:

ob\_Octo**.**eat()

Out[83]:

'campers'

**10. 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.**

In [87]:

**class** Laser:

**def** does(self):

**return** 'disintegrate'

**class** Claw:

**def** does(self):

**return** 'crush'

**class** SmartPhone:

**def** does(self):

**return** 'ring'

**class** robot():

**def** does(self):

l**=**Laser()

c**=**Claw()

s**=**SmartPhone()

**return** l**.**does(),c**.**does(),s**.**does()

In [88]:

r**=**robot()

In [89]:

r**.**does()

Out[89]:

('disintegrate', 'crush', 'ring')