Q1.Explain encapsulation and abstraction with an example.

Soln.

**Encapsulation**:

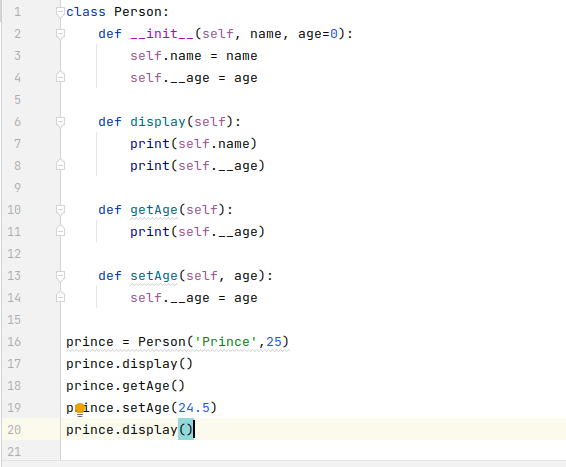
Encapsulation is the process of wrapping up variables and methods into a single entity.

**Abstraction**:

Abstraction is used to hide the irrelevant data/class in order to reduce the complexity.

In the below example we are simply calling function to print the age or name.

So user doesn't need to know about how this display function works.

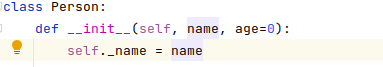


Q2.How can we define private methods and properties in a class?

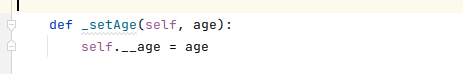
Soln.

Using ‘\_’ we can define private methods and properties in the class

Ex:



\_name(defining private property)

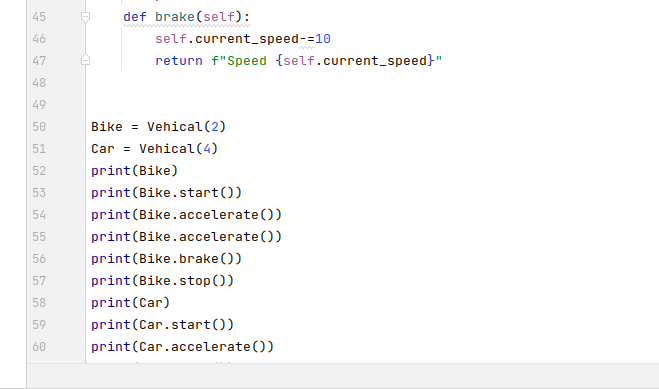


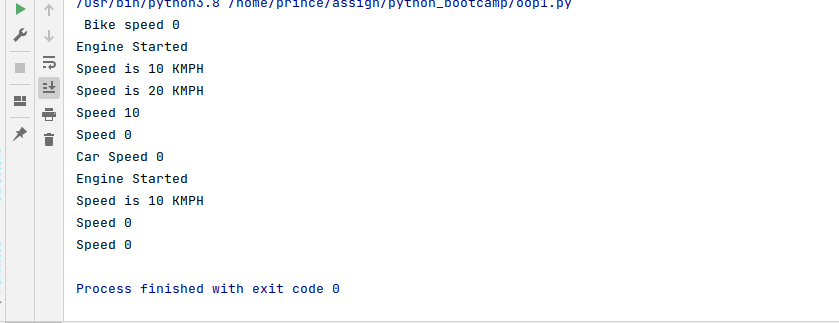
\_setAge(defining private method)

Q3.Create a class Vehicle with propertie no\_of\_wheels, current\_speed (default value should be zero). Create four functions start, stop, accelerate (which increases the current speed by 10KMPH) and break(which reduces the speed by 10KMPH). Create two objects of the class and name it car and bike.

Soln.







Q4.Create a class User with name and email properties. Create 4 objects of the class. Make sure printing the object prints the email of that user.

user1 = User('amit', 'amit.dubey@tothenew.com)

Soln.





Q5.Create a class Node that contains two properties data and next(reference to the next node). Create a class LinkedList with property head that is the first node of the list. The LinkedList shoud be instantiated as:

list\_1 = LinkedList([1,2,3,4])

print(l) printing the list should give following info

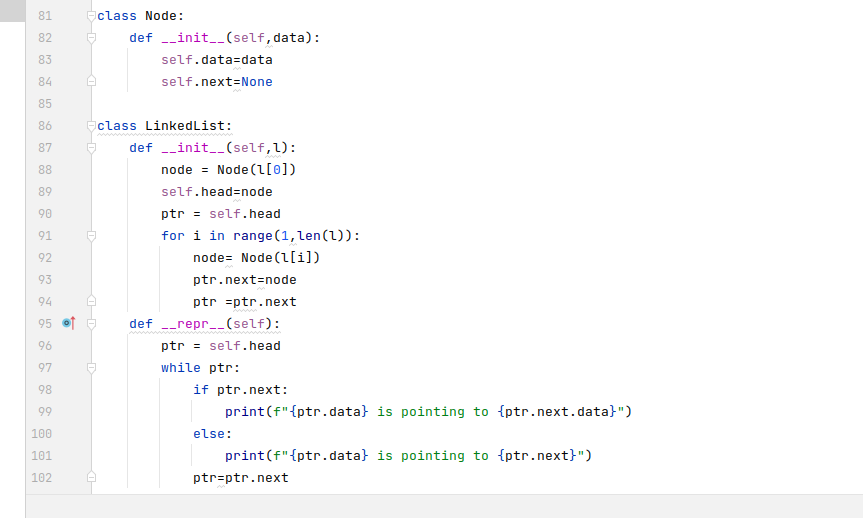
1 is pointing to 2

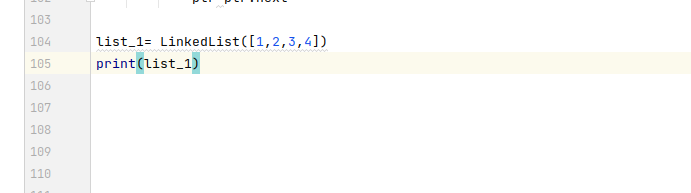
2 is pointing to 3

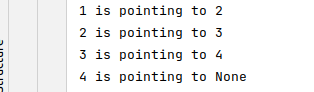
3 is pointing to 4

4 is pointing to None

Soln.

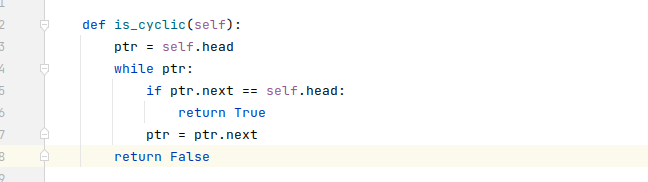


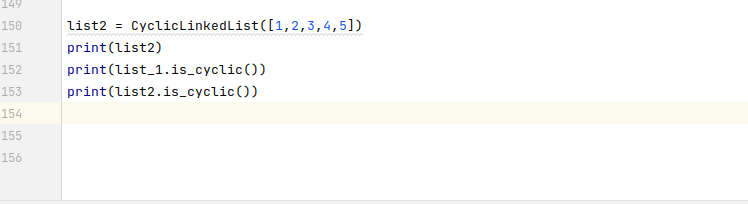




Q6.Add a function in LinkedList class 'is\_cyclic(self)' which checks if a linked list has a loop. Create a cyclic LinkedList and an acyclic list and call is\_cyclic function on both the objects.

Soln.





I have provided proper solution for this question on program file which i have uploaded on github(oop1.py)