Q1. What is the difference between \_\_getattr\_\_ and \_\_getattribute\_\_?

Ans: \_\_getattr\_\_ is implemented later if AttributeError is generated by \_\_getattribute\_\_, \_\_getattribute\_\_ is used to find an attribute of a class. It raises an AttributeError if it fails to find an attribute of a class, but \_\_getattribute\_\_ and \_\_getattr\_\_ both have to be defined in the same class. If no attribute is found, \_\_getattr\_\_ returns a default value. So the key difference is that \_\_getattr\_\_ is called for attributes that don't actually exist in a class.

Q2. What is the difference between properties and descriptors?

Ans: With Properties we can bind multiple functions together (getter, setter and delete functions) with an attribute name, using the built-in property function or @property decorator. So, each reference to an attribute looks like simple, direct access

With Descriptor we can bind getter, setter and delete functions into a seperate class. then assign an object of this class to the attribute name in our main class. So, each reference to an attribute looks like simple, direct access but invokes an appropriate function of the descriptor object.

Q3. What are the key differences in functionality between \_\_getattr\_\_ and \_\_getattribute\_\_, as well as properties and descriptors?

Ans: \_\_getattr\_\_ method will be used when we request an attribute which is not already defined

\_\_getattribute\_\_ will invoke before looking at actual attributes on the object. Python invokes this method for every attribute regardless whether it exists or not.

With Properties we can bind multiple functions together (getter, setter and delete functions) with an attribute name, using the built-in property function or @property decorator. So, each reference to an attribute looks like simple, direct access

With Descriptor we can bind getter, setter and delete functions into a seperate class. then assign an object of this class to the attribute name in our main class. So, each reference to an attribute looks like simple, direct access but invokes an appropriate function of the descriptor object.