Q1. What is the difference between __getattr__ and __getattribute__?
Answer: __getattr__ is invoked when an attribute is not found in a class. __getattribute__ is invoked to access the existing attributes of a class.

__getattr__ is not usually called unless __getattribute__ is invoked and raises an "Attribute Error"

Q2. What is the difference between properties and descriptors?
Answer: Descriptors are a low level mechanism that allow the user to hook into the attributes of an object being accessed. Properties are high level implementation of descriptors, or in other words properties are implemented using descriptors.

Descriptors are python objects that manage the attributes of different classes which use the object as reference. Descriptors use __getters__(), __setters__() and __delete() method, any object containing any of the three methods can be called a descriptor

Property is a way to encompass the get, set and delete methods and manage them internally without changing the application interface


```
Example:

class Circle:
    def __init__(self, radius):
        self._radius = radius

    def _get_radius(self):
        print("Get radius")
        return self.radius

    def _set_radius(self, value):
        print("Set radius")
        self.radius = value

    def _del_radius(self):
        print("Delete radius")
        del self.radius

    radius = property(
        fget=_get_radius,
        fset=_set_radius,
        fdel=_del_radius,
        doc="The radius property."
    )
```

Q3. What are the key differences in functionality between __getattr__ and __getattribute__, as well as properties and descriptors?

Answer: __getattribute__ is always called. Essentially this method is used to find an attribute of a class. If it fails, it will raise an AttributeError. In case it fails, and class implements __getattr__, then __getattr__ is called right after. Therefore, the biggest difference is that __getattr__ is called for attributes that don't actually exist in a class.

Descriptors are a low-level mechanism that lets you hook into an object's attributes being accessed. Properties are a high-level application of this; that is, properties are implemented using descriptors