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

The main difference between \_\_getattr\_\_ and \_\_getattribute\_\_ methods in Python lies in how they are invoked and their behavior when accessing attributes of an object.

\_\_getattr\_\_:

\_\_getattr\_\_ is called only when an attribute is not found through the usual process of attribute lookup

It is invoked when an attribute is accessed that does not exist as an instance attribute or in the class hierarchy

It takes the attribute name as a parameter and allows you to dynamically generate or retrieve the value for that attribute.

\_\_getattribute\_\_:

\_\_getattribute\_\_ is called for every attribute access, regardless of whether the attribute exists or not.

It is invoked for every attribute access, including instance attributes and those inherited from the class hierarchy.

Q2. What is the difference between properties and descriptors?

Properties: Properties are defined within a class and are specific to individual attributes of the class. They provide a way to define methods that are invoked when getting, setting, or deleting the value of an attribute

Properties: Properties are implemented using decorator syntax or by defining getter, setter, and deleter methods within a class. They provide a clean and intuitive way to encapsulate attribute access and perform additional logic when accessing the attribute.

Descriptors: Descriptors are more general and can be used across multiple attributes or classes. They are defined as separate objects and can be reused across different attributes or classes.

Descriptors: Descriptors are implemented by defining a class that implements the \_\_get\_\_, \_\_set\_\_, or \_\_delete\_\_ methods. These methods define the behavior when getting, setting, or deleting the attribute.

properties are typically used to customize attribute access for specific attributes within a class, providing a clean and straightforward way to encapsulate attribute logic. Descriptors, on the other hand, offer a more general and flexible mechanism for controlling attribute access and can be reused across multiple attributes or classes.

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

The key differences in functionality between \_\_getattr\_\_, \_\_getattribute\_\_, properties, and descriptors are as follows.

Invocation and Attribute Lookup

Error Handling

Access Control

Scope of Application

Implementation