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

Ans : \_\_getattr\_\_ :It return the value of attributes which is constructed inside the class and having an class object which store its value.If the attribute is not present inside the class and if default value is provided in the syntax only then it will return that value else it will show attribute error

\_\_getattribute\_\_ : This method will invoked before looking at the actual attributes on the object. Means, if ywe have getattribute method in our class, python invokes this method for every attribute regardless whether it exists or not.

Q2. What is the difference between properties and descriptors?

Ans . Properties serve a specific role, while descriptors are more general. Properties define

get, set, and delete functions for a specific attribute;

descriptors provide a class with methods for these actions, too, but they provide extra flexibility to

support more arbitrary actions.

In fact, properties are really a simple way to create a specific

kind of descriptor—one that runs functions on attribute accesses.

Coding differs too: a property is created with a built-in function, and a descriptor is coded with

a class; thus, descriptors can leverage all the usual OOP features of classes, such

as inheritance.

Moreover, in addition to the instance’s state information, descriptors

have local state of their own, so they can sometimes avoid name collisions in

the instance.

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

* he \_\_getattr\_\_ and \_\_getattribute\_\_ methods are more generic: they can be used

to catch arbitrarily many attributes.

* In contrast, each property or descriptor provides

access interception for only one specific attribute—we can’t catch every attribute

fetch with a single property or descriptor.

* On the other hand, properties

and descriptors handle both attribute fetch and assignment by design: \_\_get

attr\_\_ and \_\_getattribute\_\_ handle fetches only; to intercept assignments as well,

\_\_setattr\_\_ must also be coded.

* The implementation is also different: \_\_get

attr\_\_ and \_\_getattribute\_\_ are operator overloading methods, whereas properties

and descriptors are objects manually assigned to class attributes.

* Unlike the others, properties and descriptors can also sometimes avoid extra calls on

assignment to unmanaged names, and show up in dir results automatically, but are also narrower

in scope—they can’t address generic dispatch goals

.