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

**Ans: \_\_getattribute\_\_** is used to find an attribute of a class. It raises an **Attribute Error** if it fails to find an attribute of a class.

**\_\_getattr\_\_** is implemented letter if Attribute Error is generated by **\_\_getattribute\_\_,** but for this **\_\_getattribute\_\_**and **\_\_getattr\_\_** both has to be define in same class. If no attributeis found, **\_\_getattr\_\_** returns a default value. So key difference is that **\_\_getattr\_\_** is called for attributes that don’t actually exist on a class

**Q2. What is the difference between properties and descriptors?**

Ans: the difference between properties and Descriptors:

Properties : with properties we can bind getter , setter and delete function together with an attribute name , using the built-in property function or @property decorator. When we do this , each reference to an attribute looks like simple , direct access , but involves the appropriate function of the object

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

**Ans:** The key difference between \_\_getattr\_\_, \_\_getattribute\_\_, properties and Descriptors are:

**\_\_getattr\_\_ :** python will call this method whenever you request an attribute that hasn’t already been defined.

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

**Properties :** with properties we can bind getter, setter and delete function together with an attribute name, using the built-in property function or @property decorator. When we do this, each reference to an attribute looks like simple , direct access, but involves the appropriate function of the object.

**Descriptor:** with descriptor we can bid getter, setter and delete functions into a separate class. We then assign an object of this class to the attribute name in our main class .when we do this, each reference to an attribute looks like simple, direct access but invokes an appropriate function of descriptor object.