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

Ans: \_\_getattribute\_\_ has a default implementation, but \_\_getattr\_\_ does not. This has a clear meaning: since \_\_getattribute\_\_ has a default implementation, while \_\_getattr\_\_ not, clearly python encourages users to implement \_\_getattr\_\_ .

Q2. What is the difference between properties and descriptors?

Ans : The Cliff's Notes version: 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.

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

Ans : getattribute: Is used to retrieve an attribute from an instance. It captures every attempt to access an instance attribute by using dot notation or getattr() built-in function. getattr: Is executed as the last resource when attribute is not found in an object.