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

\_\_getattr\_\_ is called when you try to access an attribute that doesn't exist, allowing you to provide custom behavior for missing attributes.

\_\_getattribute\_\_ is called for all attribute accesses and is used for more general attribute interception, but it requires careful implementation to avoid issues like infinite recursion.

Q2. What is the difference between properties and descriptors?

Ans:  
Properties are a straightforward way to customize attribute access with decorators like @property.

Descriptors provide finer-grained control by defining methods like \_\_get\_\_, \_\_set\_\_, and \_\_delete\_\_ and are more versatile for attribute behavior customization.

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

- `\_\_getattr\_\_` and `\_\_getattribute\_\_` differ in when they are called: `\_\_getattr\_\_` for missing attributes, `\_\_getattribute\_\_` for all attributes.

- Properties are a simpler way to customize attribute access with decorators.

- Descriptors offer more control by defining `\_\_get\_\_`, `\_\_set\_\_`, and `\_\_delete\_\_` methods and are more versatile for attribute behavior customization.