**Python advance assignment -10**

**Q1. What is the difference between getattr and getattribute?**

**\_\_getattr\_\_** is a method that is called when an attribute is accessed and it is not found in the object's dictionary. It can be used to define a fallback behavior for attribute access, such as returning a default value or raising an exception. **\_\_getattribute\_\_** is a method that is called for every attribute access, regardless of whether the attribute is found in the object's dictionary or not. It can be used to define custom attribute access behavior for all attributes.

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

Properties are a way to define methods that are accessed like attributes. They use the **@property** and **@<property\_name>.setter** decorators to define a getter and setter method for the property. Descriptors are a more general way to define custom attribute behavior. They define methods like **\_\_get\_\_**, **\_\_set\_\_**, and **\_\_delete\_\_** to define custom behavior for getting, setting, and deleting attributes.

**Q3. What are the key differences in functionality between getattr and getattribute, as well as properties and descriptors?**

**\_\_getattr\_\_** is called when an attribute is not found in the object's dictionary, while **\_\_getattribute\_\_** is called for every attribute access. Properties are a way to define methods that are accessed like attributes and they use decorators to define a getter and setter method for the property. Descriptors are a more general way to define custom attribute behavior, they define methods like **\_\_get\_\_**, **\_\_set\_\_**, and **\_\_delete\_\_** to define custom behavior for getting, setting, and deleting attributes. Properties are simpler and can be used for simple use cases, while descriptors are more powerful and flexible.