1. State the difference between constructor and destructor in python.

Answer :-

**Constructor**:

- A constructor in Python is a special method that is automatically called when an object of a class is created.
- It is typically used to initialize the attributes of the class.
- The constructor method in Python is named `__init__()`.

**Destructor**:

- A destructor is a method that is automatically called when an object is about to be destroyed or when it goes out of scope.
- It is used to perform cleanup operations like closing files or releasing resources.
- The destructor method in Python is named `__del__()`.
-

2. State the difference between Public and Private access modifier in python.

Answer :-

- **Public Access Modifier**:

  - Public access allows class attributes and methods to be accessed from anywhere, both within and outside the class.
  - In Python, class members (attributes and methods) are public by default unless explicitly made private or protected.
  - This makes it easy to interact with objects, but can make the internal workings of the class less secure as everything is accessible.

- **Private Access Modifier**:

  - Private access restricts the visibility of attributes and methods, allowing them to be accessed only within the class they are defined in.
  - It is used to protect the internal state of an object, ensuring that data or methods are not accidentally altered or accessed from outside the class.

- In Python, attributes or methods can be made private by prefixing their names with a double underscore (__).

3. Explain Method overriding in python Inheritance.

Answer :-

- **Method overriding** is a concept in object-oriented programming where a subclass provides its own implementation of a method that is already defined in its superclass.
- This allows the subclass to have a specialized behavior for a method while keeping the same method name and signature as the parent class.
- When a method is overridden, calling the method on an instance of the subclass will invoke the subclass's version of the method, even if the method is originally defined in the parent class.
- Method overriding is crucial for achieving **polymorphism**, where a single interface can be used to represent different underlying forms (data types).
- It helps in situations where the behavior of the method needs to change for a specific subclass, providing more flexibility and control over how methods work in inherited classes.

**4. What is Abstraction in OOPS and How Does It Simplify Complex Systems?**

- **Abstraction** is a fundamental concept in object-oriented programming (OOP) that focuses on exposing only the essential features of an object while hiding the complex implementation details.
- It allows developers to define a simplified view of an object, where the internal logic and processes are hidden from the user. The user interacts only with the object's properties and methods that are relevant to them.
- By using abstraction, the complexities and intricate details of how a class operates are kept away from the user, enabling them to focus only on how to use the object rather than how it works internally.
-