OBJECT ORIENTED PROGRAMMING :

Q1. What are global, protected and private attributes?

Ans. *Global, protected, and private attributes are access modifiers used to control access to class attributes in Python. A global attribute is accessible from anywhere in the code, a protected attribute can only be accessed within the class and its subclasses, and a private attribute can only be accessed within the class that defines it.*

Q2. What is the use of self in Python?

Ans. *In Python, self is a reference to the current instance of a class. It is used to access the attributes and methods of an object within the class definition.*

Q3. Are access specifiers used in python?

Ans. *Access specifiers such as private, protected, and public are not strictly enforced in Python, but conventionally used to indicate the intended access level of class attributes.*

Q4. Is it possible to call parent class without its instance creation?

Ans. *Yes, it is possible to call a parent class without creating an instance of it using the super() function.*

Q5. How is an empty class created in python?

Ans*. An empty class can be created in Python by simply defining a class with a pass statement.*

For example:

```
class EmptyClass:
    pass
```

Q6. How will you check if a class is a child of another class?

Ans. *To check if a class is a child of another class, you can use the issubclass() method.*

For example:

```
class ParentClass:
    pass
```

```
class ChildClass(ParentClass):
    pass
print(issubclass(ChildClass, ParentClass)) # True
```

Q7. What is docstring in Python?

Ans. *A docstring in Python is a string literal that is used to provide documentation for a function, module, class, or method. It is enclosed in triple quotes and appears as the first line of the object's definition.*

Q8. Is Python Object-oriented or Functional Programming?

Ans. *Python is an object-oriented programming language that supports both object-oriented and functional programming paradigms.*

Q9. What does an object() do?

Ans. *The object() function in Python creates a new object of the base class. It can be used to test the behavior of an object in the absence of a specific class.*

Q10. What is the purpose of the super function in inheritance, and how is it used?

Ans*. The super() function in Python is used to call a method of a parent class from a subclass. It is typically used in the implementation of inheritance to access the parent class's methods and properties. The super() function is used in the following way:*

```
class ChildClass(ParentClass):
    def some_method(self):
        super().some_method()
```

Q11. What is data abstraction?

Ans*. Data abstraction is the process of representing complex data types in a simplified and abstract manner. It is used to hide the implementation details of an object and provide a clean interface for interacting with it. In Python, data abstraction is achieved through the use of encapsulation and abstraction mechanisms such as private and protected attributes, abstract classes, and interfaces.*