**ADITYA AMIN**

**ASSIGN : 01**

Q1. What is the purpose of Python's OOP?

The purpose of Python's Object-Oriented Programming (OOP) is to provide a way to organize and structure code so that it is more modular, reusable, and easier to maintain.

Q2. Where does an inheritance search look for an attribute?

When you use inheritance in Python, if an attribute or method is not found in an object's class, Python will search for it in the object's parent class, and then in the parent class's parent class, and so on, following the inheritance chain until it finds the attribute or method or reaches the top of the inheritance hierarchy. This process is called attribute lookup.

Q3. How do you distinguish between a class object and an instance object?

In Python, a class object is the object that defines a class, while an instance object is an object that is created from a class.

Q4. What makes the first argument in a class’s method function special?

In Python, the first argument in a class's method function is conventionally named self and it refers to the instance of the class that the method is called on.

This first argument is special because it allows the method to access the instance variables and other attributes of the object that it belongs to.

Q5. What is the purpose of the \_\_init\_\_ method?

The \_\_init\_\_ method is a special method in Python classes that is called when an instance of the class is created. Its purpose is to initialize the attributes of the instance with any arguments that are passed in.

Q6. What is the process for creating a class instance?

To create an instance of a class in Python, you need to follow these steps:

Define the class: First, define the class using the class keyword, specifying any instance variables, class variables, and methods that the class will have.

Initialize the instance variables: If your class has instance variables that need to be initialized, define an \_\_init\_\_ method that takes the self argument and initializes the instance variables using the self attribute.

Create the instance: To create an instance of the class, call the class like a function and pass in any necessary arguments to the \_\_init\_\_ method.

Q7. What is the process for creating a class?

To create a class in Python, you need to follow these steps:

Use the class keyword to define the class: Start by using the class keyword followed by the name of the class you want to create.

Define the class properties: Inside the class definition, you can define any properties that you want the class to have, including instance variables, class variables, and methods.

Define the class methods: Methods are functions defined inside a class that can access the class and its instances' properties. Methods take the self parameter as the first argument, which refers to the instance of the class the method is being called on.

Q8. How would you define the superclasses of a class?

In object-oriented programming, a class can inherit attributes and behavior from one or more parent classes, also known as superclasses.

To define the superclasses of a class in Python, you need to include the superclass name inside the parentheses after the class name in the class definition, like this:

class ChildClass(SuperClass):

# class body