ADVANCE ASSIGNMENT 1

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

Ans1- The purpose of Python's Object-Oriented Programming (OOP) is to organize and structure code by modeling real-world entities as objects with attributes (data) and methods (functions). This approach promotes modularity, reusability, and maintainability of code, making it easier to manage and scale complex software projects. OOP encourages the encapsulation of data and behavior within objects, enabling abstraction to hide implementation details while exposing a clean interface.

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

Ans2-Inheritance search in Python looks for an attribute by first checking the instance itself. If the attribute is not found there, it proceeds to search the class of the instance.

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

Ans3- A class object represents the blueprint or template for creating instances. It defines the structure and behavior that instances will inherit. Class objects are created using the class keyword and store class-level attributes and methods.

An instance object, on the other hand, is a specific realization of a class, created by calling the class as if it were a function. Instances contain unique data and can access both instance-specific and class-level attributes and methods. They represent individual objects with their own state, distinct from other instances of the same class.

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

Ans4- The first argument in a class's method function, conventionally named "self," refers to the instance calling the method. It allows the method to access and manipulate the instance's attributes and state, distinguishing instance-specific data from class-level data within the method.

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

Ans5- The purpose of the \_\_init\_\_ method in Python is to initialize and set the initial state of an object when it is created.

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

Ans6- Define a class with the class keyword, including an \_\_init\_\_ method to initialize its attributes.

Instantiate an object from the class by calling the class as if it were a function, passing any required arguments to the \_\_init\_\_ method.

This call to the class constructor creates a unique instance of the class, initializing its attributes as specified in the \_\_init\_\_ method.

You can then access and manipulate the instance's attributes and methods using the instance variable created in step 2.

Q7. What is the process for creating a class?

Ans7- Use the class keyword followed by the class name to define the class.

Inside the class, define attributes and methods that describe its behavior.

Optionally, include a constructor method \_\_init\_\_ to initialize instance-specific attributes.

Instantiate objects from the class by calling the class name as a function.

Access and manipulate object attributes and methods using dot notation

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

Ans8- The superclasses of a class are the classes from which it directly inherits, specified in the class definition using parentheses after the class name.