ASSIGNMENT - 1

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

Ans: Object-oriented programming in Python allows for organizing code into objects that contain both data and methods that operate on that data. Its primary purposes include encapsulation, inheritance, and polymorphism, which help in structuring and organizing code for better reusability and maintainability.

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

Ans: When Python looks for an attribute in a class hierarchy due to inheritance, it starts searching from the current class and moves up the inheritance chain until it finds the attribute. This is known as the Method Resolution Order (MRO).

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

Ans: A class object is the blueprint/template that defines the attributes and behaviors of its instances. An instance object, on the other hand, is a specific realization of that class. For example, a class "Car" might define attributes like "color" and methods like "drive", while an instance would be a specific car object, such as "my\_car = Car()" which has its own color and can execute the drive method.

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

Ans: In Python, the first argument in a class method (conventionally named self) refers to the instance of the class on which the method is being called. It allows access to the instance's attributes and methods within that method.

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

Ans: The \_\_init\_\_ method is a special method in Python classes used to initialize object attributes. It gets called automatically when an instance of the class is created and allows setting initial values for object attributes.

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

Ans: To create an instance of a class in Python, you use the class name followed by parentheses. For example, if you have a class named Car, you create an instance like this: my\_car = Car(). This calls the \_\_init\_\_ method of the class, initializing the instance.

Q7. What is the process for creating a class?

Ans: To create a class in Python, you use the class keyword followed by the class name and a colon. Inside the class, you define attributes and methods that describe the behavior and characteristics of instances created from that class.

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

Ans: Superclasses, also known as parent classes or base classes, are the classes from which a subclass inherits. In Python, you can define a class that inherits from one or more superclasses by listing them in parentheses after the class name. For instance, class SubClass(SuperClass1, SuperClass2):.