**Python Advanced Assignment 1**

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

Ans-) The purpose of Python's Object-Oriented Programming (OOP) is to create reusable code and reduce the complexity of code by breaking it down into smaller, self-contained, and reusable components called objects. OOP allows users to create classes, which are the blueprint for objects, and define attributes (data) and methods (functions) that can be used to manipulate the data.

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

Ans-) When an attribute is accessed in a subclass, Python first looks for the attribute in the subclass itself. If the attribute is not found in the subclass, Python then searches for the attribute in its superclass. This search continues up the inheritance hierarchy until the attribute is found or until the top of the hierarchy (object) is reached.

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

Ans-) A class object is an object that is an instance of the class type. It is created when the class is defined and has access to the class's attributes and methods. An instance object, on the other hand, is created when the class is instantiated (i.e., when an object of the class is created using the class constructor). The instance object has its own set of attributes and methods that may differ from the class attributes and methods.

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

Ans-) The first argument in a class's method function is typically referred to as the "self" parameter. It refers to the instance of the class that the method is being called on. This parameter is automatically passed by Python when a method is called, and it allows the method to access and manipulate the instance's attributes.

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

Ans-) The init method is a special method in Python that is used to initialize the attributes of an object when it is instantiated. It is called automatically when an instance of the class is created and allows the user to set the initial state of the object.

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

Ans-) The process for creating a class instance involves using the class constructor to create a new object of that class type. The constructor is called by using the class name followed by parentheses (). Any arguments that the constructor requires are passed within the parentheses. For example, if the class is named MyClass, an instance of the class can be created by calling MyClass().

Q7. What is the process for creating a class?

Ans-) The process for creating a class involves defining the class using the class keyword, followed by the name of the class. Within the class definition, the attributes and methods of the class are defined. Once the class is defined, it can be instantiated to create objects of that class.

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

Ans-) The superclasses of a class are the classes from which the current class inherits attributes and methods. In Python, a class can inherit from multiple superclasses by specifying them in a comma-separated list within parentheses after the class name. For example, if a class named MyClass inherits from two superclasses named SuperClass1 and SuperClass2, the class definition would begin like this:

class MyClass(SuperClass1, SuperClass2):

# class attributes and methods go here