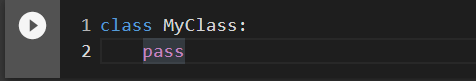
**Python Advanced Assignment 2**

Q1. What is the relationship between classes and modules?

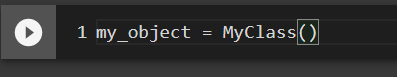
Ans-) A class is a blueprint for creating objects in Python, while a module is a file containing Python definitions and statements. A module can contain one or more classes, as well as other functions and variables. In this sense, classes and modules are related in that a class can be defined in a module, and a module can contain one or more classes.

Q2. How do you make instances and classes?

Ans-) To create a class in Python, you use the class keyword followed by the name of the class and a colon, like this:

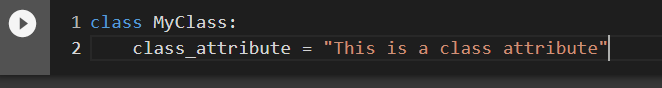


To create an instance of a class, you use the class name followed by parentheses, like this:



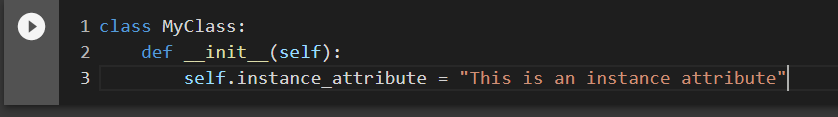
Q3. Where and how should be class attributes created?

Ans-) Class attributes should be created inside the class definition but outside of any method, like this:



Q4. Where and how are instance attributes created?

Ans-) Instance attributes are created inside the class's methods, typically inside the \_\_init\_\_() method, like this:



Q5. What does the term “self”; in a Python class mean?

Ans-) In Python, self refers to the instance of the class that is being operated on. It is the first parameter that is passed to a class method, and it allows you to access and modify instance attributes and call other instance methods.

Q6. How does a Python class handle operator overloading?

Ans-) Python classes can overload operators by defining special methods that correspond to each operator. For example, the + operator can be overloaded by defining the \_\_add\_\_() method in the class.

Q7. When do you consider allowing operator overloading of your classes?

Ans-) Operator overloading can be useful when you want to provide a more intuitive syntax for working with your objects. For example, if you have a complex number class, you might want to overload the + operator to allow users to add complex numbers using the familiar + symbol.

Q8. What is the most popular form of operator overloading?

Ans-) The most popular form of operator overloading is probably arithmetic operator overloading, such as overloading the +, -, \*, and / operators.

Q9. What are the two most important concepts to grasp in order to comprehend Python OOP code?

Ans-) The two most important concepts in Python OOP code are classes and objects. Classes define the blueprint for objects, while objects are instances of classes that can be manipulated and interacted with in various ways. Additionally, the concept of inheritance is also important in Python OOP, as it allows you to create new classes based on existing ones and reuse code.