Python Advance – 2

1. What is the relationship between classes and modules?

**The difference between a class and a module in python is that a class is used to define a blueprint for a given object, whereas a module is used to reuse a given piece of code inside another program.**

A class can have its own instance, but a module cannot be instantiated. We use the ‘class’ keyword to define a class, whereas to use modules, we use the ‘import’ keyword. We can inherit a particular class and modify it using inheritance. But while using modules, it is simply a code containing variables, functions, and classes.Modules are files present inside a package, whereas a class is used to encapsulate data and functions together inside the same unit.

2. ) How do you make instances and classes?

**Use the class name to create a new instance**

Call ClassName() to create a new instance of the class ClassName . To pass parameters to the class instance, the class must have an \_\_init\_\_() method. Pass the parameters in the constructor of the class.

1. ) Where and how should be class attributes created?

Instance attributes are attributes or properties attached to an instance of a class. Instance attributes are **defined in the constructor**. Defined directly inside a class. Defined inside a constructor using the self parameter.

4.) Where and how are instance attributes created?

Instance attributes are defined **in the constructor**. Defined directly inside a class. Defined inside a constructor using the self parameter. We add instance attributes usually by defining the constructor method “**\_\_init\_\_”**.

5.) What does the term “self” in a Python class mean?

self represents the instance of the class. By using the “self”  we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.  
The reason you need to use self. is because Python does not use the syntax to refer to instance attributes. Python decided to do methods in a way that makes the instance to which the method belongs be passed automatically, but not received automatically: the first parameter of methods is the instance the method is called on.

*Self is always pointing to Current Object.*

6.) How does a Python class handle operator overloading?

Operator Overloading means giving extended meaning beyond their predefined operational meaning. For example operator + is used to add two integers as well as join two strings and merge two lists. It is achievable because ‘+’ operator is overloaded by int class and str class. You might have noticed that the same built-in operator or function shows different behavior for objects of different classes, this is called *Operator Overloading*.

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

Operator Overloading means **giving extended meaning beyond their predefined operational meaning**. For example operator + is used to add two integers as well as join two strings and merge two lists. It is achievable because '+' operator is overloaded by int class and str class.

8.) What is the most popular form of operator overloading?

A very popular and convenient example is the **Addition (+) operator**. Just think how the '+' operator operates on two numbers and the same operator operates on two strings. It performs “Addition” on numbers whereas it performs “Concatenation” on strings.

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

Two key concepts of OOP which are **inheritance and polymorphism**. Both inheritance and polymorphism are key ingredients for designing robust, flexible, and easy-to-maintain software.