**Answer 1:**

A module can have multiple classes in a .py files and classes is just a blueprint for creating object.

**Answer 2:**

We can instantiate the object by directly calling the class name with parenthesis

Class A:

print(“Hello this is A class”)

objA=A() <<- This is how to instantiate the object

**Answer 3:**

Class Attributes are created within the class namespace.

**Answer 4:**

Instance attributes are personal to the instances, Every object has its own copy of the instance attribute.

**Answer 5:**

Self keyword is used to represent an instance(object) of the given class, It stores the reference of the object.

**Answer 6:**

As the operator are overloaded with more than one classes, Example “+” operator can be used to add two integers and two string with the help of int class and str class.

**Answer 7:**

Python provides some special function or magic function to perform operator overloading

Example: We have one magic function \_\_add\_\_ which is automatically invoked in which operation for + operator is defined.

**Answer 8:**

A very popular and convenient example is the **Addition (+) operator**.

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.

Operators in Python work for **built**-**in** classes, like **int**, **str**, **list**, etc. But you can extend their **operability** such that they work on objects of user-defined classes too.

**Answer 9:**

We should know about Polymorphism and Inheritance.