Q1. Define the relationship between a class and its instances. Is it a one-to-one or a one-to-many partnership, for example?

Answer: Relationship between classes is one to one if one to one if only object is created for a particular class but it can also handle one to many relations if we create many objects of the same classes.

Q2. What kind of data is held only in an instance?

Answer: An instance of a class holds variable data which is specific to a particular object of a specific class.

Q3. What kind of knowledge is stored in a class?

Answer: An object can store anything specific to a class such as variables, class specific functions and they can also be used to return specific value after some values are being passed to them and they do return some value after computation.

Q4. What exactly is a method, and how is it different from a regular function?

Answer: Methods can be called on an object and can alter the object’s state which a function cannot do .

Q5. Is inheritance supported in Python, and if so, what is the syntax?

Answer:

class Person:  
  def \_\_init\_\_(self, fname, lname):  
    self.firstname = fname  
    self.lastname = lname  
  
  def printname(self):  
    print(self.firstname, self.lastname)  
  
x = Person("John", "Doe")  
x.printname()

class Student(Person):

# this part is called inheritance as student is inheriting from Person and will the values from parent class as well

  def \_\_init\_\_(self, fname, lname):  
    Person.\_\_init\_\_(self, fname, lname)

Q6. How much encapsulation (making instance or class variables private) does Python support?

Answer: We can make a class variable private as adding double underscores before class variables example:

\_\_var1 = some value

\_\_var2 = another value

Q7. How do you distinguish between a class variable and an instance variable?

Answer: We cannot directly access a class variable. An instance variable is one which is declared for passing values to a class and we can also access a class variable with help of an instance variable as well.

Q8. When, if ever, can self be included in a class's method definitions?

Answer: If we want to assign a class method’s variable a particular value, then we can use self.variablename = value to assign a new value of overwrite the old value.

Q9. What is the difference between the \_ \_add\_ \_ and the \_ \_radd\_ \_ methods?

Answer: We results of both \_\_add\_\_ and \_\_radd\_\_ methods are equivalent , it only depends on how you are using operands to get your results.

Q10. When is it necessary to use a reflection method? When do you not need it, even though you support the operation in question?

Answer: Reflection methods are used when we need to examine the attributes about objects that might be passed as parameters to a function.

Q11. What is the \_ \_iadd\_ \_ method called?

Answer: \_\_idadd\_\_ nethod is used to add a new value to old value by overwriting a new value to old value after addition is made in this function.

Q12. Is the \_ \_init\_ \_ method inherited by subclasses? What do you do if you need to customize its behavior within a subclass?

Answer: \_\_init\_\_() is used to get all or some of the class variables from parent class which are inherited by a child class.