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

Ans:  **the relationship between a class and its instance is one to many meaning one class has different instances.**

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

Ans: **instance variables hold data that must be referenced by one or more than one method of that instance class such as constructor.**

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

Ans: **classes can have attributes, constructor, and methods.**

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

Ans: **A function is called method if it written inside a class. Regular function can be called independently and no instance needs to be created for that but for a class method, an instance of the class needs to be created first to be able to call the method.**

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

Ans: **yes, it is supported, for inheriting the super class, a subclass parameter needs to be passed as a superclass name so that all of the superclass methods can be inherited.**

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

Ans: **you can make as many instances of a class an d private variables as you want but private variables can be accessed outside the class provided that the user knows the method and the variable name.**

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

Ans: **instance variable values vary from object to object and a class variable is declared inside a class but outside the constructor or any other method.**

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

Ans: ‘**self’ as a pointer has to be included all the time in a class method definition to allow an instance to access the class method.**

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

Ans: **x.\_\_add\_\_(y) is used for addition and x.\_\_radd\_\_(y) is used for right addition. The former calculates x+y and the latter calculates y+x**

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

Ans: **you need to use reflection when you need to use the reflection-enabling function such as type(), isinstance(), getatttr(). It is necessary to use it when you want to write recursive reverse functions.**

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

Ans: **This magic method is used to perform in place operation, meaning it adds the operands and assigns the value to the left operand.**

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

Ans: **yes, by default the \_\_init\_\_ method is inherited by a subclass. You can use the super() method to call the superclass init method and modify it.**