**Assignment\_19**

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: Class is a blueprint that is needed to make an object (=instance).The difference between an object and an instance is, an object is a thing and an instance is a relation. Instance describes the relation of an object to the class that the object was made from.**

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

**Ans: The data which class needs only stored in an instance. It usually reserves memory for data that the class needs.**

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

**Ans: class is basically the blue print of data member and function.**

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

**Ans: Methods are associated with the objects of the class they belong to. Functions are not associated with any object**.

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

**Ans: Yes, inheritance is supported in python. Parent class is the class being inherited from, also called base class or child class.**

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

**Ans: There is not as encapsulation in python, because python believes the user who is writing the code is matured enough to encapsulate his code accordingly.**

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

**Ans: Instance Variable reserves memory for data that the class needs and and class variable maintains a single shared value for all instance of class even if no instance object of the class exists.**

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

**Ans: In every method, self will act as a pointer to point the data.**

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

**Ans: \_\_radd\_\_ is only called if the left object does not have an \_\_add\_\_ method, or that method does not know how to add the two object.**

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: Reflection is used while writing a framework that has to interoperate with user-defined classes, where the framework author doesn’t know what the members will be.**

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

**Ans: \_\_add\_\_ method is used to add the attributes of the class instance. For ex- let’s say object1 is an instance of a class A object2 is an instance of class B and both of these classes have an attribute called ‘a’, that holds an integer, then use \_\_iadd\_\_ it.**

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

**Ans: Yes, the \_\_init\_\_ method is inherited by subclass.**