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 : A 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.

* HOD is a staff member of college.
* All teachers are staff member of college.
* HOD and teachers has id card to enter into college.
* HOD has a staff that work according the instruction of him.
* HOD has responsibility to undertake the works of teacher to cover the course in fixed time period.

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

Ans : An instance variable is a variable which is declared in a class but outside of constructors, methods, or blocks. Instance variables are created when an object ..

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

Ans : Individually, each knowledge class is an organized entity dynamically linked to web information resources. Collectively, this kind of knowledge ...

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

Ans : A function doesn't need any object and is independent, while the method is a function, which is linked with any object. We can directly call the function with its name, while the method is called by the object's name. Function is used to pass or return the data, while the method operates the data in a class.

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

Ans : Is inheritance possible in Python? Answer. An object-oriented programming language like Python, not only supports inheritance but multiple inheritance as well. The mechanism of inheritance allows programmers to create a new class from a pre-existing class, which supports code reusability.

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

Ans : Encapsulation can be achieved by declaring the data members and methods of a class either as private or protected. But In Python, we don't have direct access modifiers like public, private, and protected. We can achieve this by using single underscore and double underscores.

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

Ans :

|  |  |
| --- | --- |
| Instance Variable | Class Variable |
| It usually reserves memory for data that the class needs. | It usually maintains a single shared value for all instances 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 : The self is used to represent the instance of the class. With this keyword, you can access the attributes and methods of the class in python. It ..

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

Ans : There is no difference. One performs a+b , the other b+a . Likewise rmul and mul produce the same result, but rsub and sub & rdiv and div do not.

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 allows programmer to access entities in program dynamically. i.e. while coding an application if programmer is unaware about a class or its methods, he can make use of such class dynamically (at run time) by using reflection.

You can use reflection to dynamically create an instance of a type, bind the type to an existing object, or get the type from an existing object and invoke its methods or access its fields and properties. If you are using attributes in your code, reflection enables you to access them.

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

Ans : Syntax. What is this? The Python \_\_iadd\_\_() magic method implements in-place addition x += y that adds together the operands and assigns the result to the left operand. This operation is also called augmented arithmetic assignment.

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

Ans : It is called as a constructor in object oriented terminology. This method is called when an object is created from a class and it allows the class to initialize the attributes of the class.