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

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

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

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

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

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

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

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

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

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

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

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

Answer

Q1. A class is a blueprint or template for creating objects, while an instance is an individual object created from a class. It is a one-to-many relationship, as one class can create many instances.

Q2. Instance variables hold data that is unique to each instance of a class.

Q3. A class can store knowledge or data that is shared among all instances of the class. This includes class variables and methods.

Q4. A method is a function that is defined inside a class and is used to define the behavior of the instances of the class. It is different from a regular function in that it has access to the instance and class variables, and is invoked on an instance of the class.

Q5. Yes, inheritance is supported in Python. The syntax for defining a subclass that inherits from a superclass is as follows:

class Subclass(Superclass):

# subclass methods and variables

Q6. Python supports encapsulation to some extent, by using underscores to indicate private variables and methods. However, these are only conventions and can still be accessed from outside the class.

Q7. A class variable is a variable that is shared by all instances of a class, while an instance variable is unique to each instance. A class variable is defined inside the class but outside of any method, while an instance variable is defined inside the constructor method of the class.

Q8. The self parameter is included in a class's method definitions to reference the instance of the class on which the method is being called. It is necessary to include self in every method definition that will be called on an instance of the class.

Q9. The **\_\_add\_\_** method is used to implement the addition operation, while the **\_\_radd\_\_** method is used to implement addition when the left operand does not support addition with the right operand.

Q10. A reflection method is used to access or modify an object's attributes at runtime. It is necessary when the attribute name is not known until runtime. However, if the attribute name is known at compile time, reflection is not necessary and the attribute can be accessed directly.

Q11. The **\_\_iadd\_\_** method is called for in-place addition using the **+=** operator.

Q12. The **\_\_init\_\_** method is inherited by subclasses, and can be customized within a subclass by defining a new **\_\_init\_\_** method that overrides the parent class's **\_\_init\_\_** method. To customize the behavior of the **\_\_init\_\_** method without completely overriding it, the **super()** function can be used to call the parent class's **\_\_init\_\_** method.