

1. What is the relationship between classes and modules?

Ans: class is used to define a blueprint for a given object and a module is used to reuse a given piece of code inside another program.

2. How do you make instances and classes?

Ans: To create a class: use the class keyword followed by classname and semicolon.

Example: class Tree:

```
def __init__(self, name):  
    self.name = name
```

To create a class instance: call a class by its name and pass the arguments which its `__init__` method accepts.

Example: fruit = Tree('Mango')

Here fruit is an instance of class Tree with argument name 'Mango'.

3. Where and how should be class attributes created?

Ans: Class attributes are created in the top of class definition outside all methods.

4. Where and how are instance attributes created?

Ans: Instance attributes are defined within the `__init__` method of class

5. What does the term 'self' in a Python class mean?

Ans: self represents the instance of the class . By using the "self" keyword we can access the attributes and methods of the class within the class in python. It binds the attributes with the given arguments.

6. How does a Python class handle operator overloading?

Ans: Python Classes handle operator overloading by using special methods called Magic methods.

7. When do you consider allowing operator overloading of your classes?

Ans: When we want to have different meaning for the same operator according to the context.

8. What is the most popular form of operator overloading?

Ans: the + operator will perform arithmetic addition on two numbers, merge two lists, or concatenate two strings

9. What are the two most important concepts to grasp in order to comprehend Python OOP code?

Ans: Classes and objects