Q1. What is the relationship between classes and modules?

Ans: Modules are collections of methods and constants. They cannot generate instances. Classes may generate instances (objects), and have per-instance state (instance variables).

Q2. How do you make instances and classes?

Ans: call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts

Q3. Where and how should be class attributes created?

Ans: A class attribute is shared by all instances of the class. To define a class attribute, you place it outside of the \_\_init\_\_() method. Use class\_name. class\_attribute or object\_name.

Q4. Where and how are instance attributes created?

Ans: Instance attributes are attributes or properties attached to an instance of a class. Instance attributes are defined in the constructor. Defined directly inside a class. Defined inside a constructor using the self parameter

Q5. What does the term "self" in a Python class mean?

Ans: self represents the instance of the class.

Q6. How does a Python class handle operator overloading?

Ans: Python provides some special function or magic function that is automatically invoked when it is associated with that particular operator.

Q7. When do you consider allowing operator overloading of your classes?

Ans: Operator overloading allows C/C++ operators to have user-defined meanings on user-defined types (classes).

Q8. What is the most popular form of operator overloading?

Ans: Addition (+) operator.

Q9. What are the two most important concepts to grasp in order to comprehend Python OOP code?

Ans: inheritance and polymorphism are fundamental concepts of object oriented programming.