Q1. What is the relationship between classes and modules?

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?

Classes are created by keyword class. o create instances of a class, you call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts

Q3. Where and how should be class attributes created?

Class attributes are created within the class definition, but outside of any methods.

Q4. Where and how are instance attributes created?

Instance attributes are created **in the constructor of a class**. The constructor is a special method that is called when an object is created. The constructor takes the new object as an argument, and it can be used to initialize the object's attributes.

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

The self keyword is used to represent an instance (object) of the given class.

Q6. How does a Python class handle operator overloading?

Operator overloading in Python allows programmers to change the behavior of built-in operators when used with user-defined objects. For example, you can define a class that overloads the + operator so that it adds two objects of the class together.

To overload an operator, you need to define a special method that corresponds to the operator. For example, to overload the + operator, you would define a method called \_\_add\_\_. The \_\_add\_\_ method takes two objects of the class as its arguments and returns the result of the operation.

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

Operator overloading is a feature in programming languages that allows the meaning of an operator to be changed **when it is used with user-defined classes**. This can be used to make your custom classes work with familiar syntax, making code more intuitive and easier to read.

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

The most common use of the addition operator '+' is for concatenating and combining two different strings

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

Both inheritance and polymorphism are fundamental concepts of object oriented programming. These concepts help us to create code that can be extended and easily maintainable