Q1. What is the relationship between classes and modules?

**Ans**. Modules are files present inside a package, whereas a class is used to encapsulate data and functions together inside the same unit.

Q2. How do you make instances and classes?

**Ans.** To create instances of a class, you call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts and class is create with simple class function such as Class player

Q3. Where and how should be class attributes created?

**Ans.** Class attributes are the variables defined directly in the class that are shared by all objects of the class.

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. By using the “self” we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

Q6. How does a Python class handle operator overloading?

**Ans.** Python operators work for built-in classes. But the same operator behaves differently with different types. For example, the + operator will perform arithmetic addition on two numbers, merge two lists, or concatenate two strings

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

**Ans.** When we have to provide an intuitive interface to users of your class, plus makes it possible for templates to work equally well with classes and built-in/intrinsic types.

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**.polymorphism and inheritance.