Q1. What is the relationship between classes and modules?

Classes are blueprint of how data and its behaviors needs to be structured

and associated together.

Modules are files which contain classes, functions and other statements which are normally grouped together

so that it can be imported all together.

Q2. How do you make instances and classes?

Classes are made using class keyword followed by its name.

Instances are made using asn assigment operation where instance name is given to the

object returned from the class invocation.

Q3. Where and how should be class attributes created?

Class attributes should be created within the class definition.

Q4. Where and how are instance attributes created?

Instance attributes are created during the creation of instance within the initiation function.

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

"self" is reference to the instance object from which the function gets called.

Q6. How does a Python class handle operator overloading?

Depending on the type of the object as operands, operator could exhibit different behavior.

That is called operator overloading.

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

Operator overloading could be used when you need to use builtin python functions along with the

objects of a user defined class.

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

The most popular operator overloading is for '+' operator where it can be used to add

int, float, string, list.

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

Inheritance and polymorphism.