Q1. What is the relationship between classes and modules?

A module can have none or multiple classed implemented inside it. Essentially modules are the .py files that contain the class definitions and other code. Modules can also be imported into other modules and programs.

Q2. How do you make instances and classes?

To create an instance of a class we call the class using the classname and pass the arguments required by it as specified in the \_\_init\_\_() method of the class.

classname(\*args)

Q3. Where and how should be class attributes created?

Class attributes are declared inside the class body and outside the functions defined in the class

class Classname:

attr1 = value1

attr2 = value2

attr3 = value3

def function1():

……

……

……

Q4. Where and how are instance attributes created?

Instance attributes are declared inside the instance body

def function1():

attr1 = value1

attr2 = value2

attr3 = value3

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

“self ” is use in a class to create an instance of the class itself while writing methods in a class. It provides access to the attributes of the class to the method.

The word “self” itself doesn’t mean anything in python, it is not a keyword. It is however seen as a general practice to use self instead of anything else to avoid any confusion that may arise.

Q6. How does a Python class handle operator overloading?

If operator overloading is used in a class then python will use the new definitions of the operator provided in the class rather than the conventional inbuilt definition for what the operator does.

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

Operator overloading can be considered when we may want to describe what we want an operator to do for a certain user defined data type.

For example:

A shopping app or website uses a “cart” to keep track of the items and their number that a customer will want to buy. The + operator in this case can be use to always return the total number of items when using the cart objects datatype.

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

The + operator is used in different ways for different datatypes

For Example:

For numeric datatypes it adds the numbers

For string type it concatenates the string

For Boolean datatypes it processes the values logically

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

Inheritance and polymorphism are the two important concepts to understand a python OOP code.

These fundamental concepts are used widely to increase productivity, flexibility and reusability of code