**Assignment \_2**

Q1. What is the relationship between classes and modules?

**Ans: Class is blueprint of a given object and they generate instances (objects).**

**Module are collection of methods and constants and cannot generate instances.**

Q2. How do you make instances and classes?

**Ans: To create instance of a class, call the class using name and pass the objects u want to create in its \_init\_ method.**

**And to create class , use the keyword “class class\_name”.**

Q3. Where and how should be class attributes created?

**Ans: Class attributes belongs to class itself they will be shared by all the instance.**

Q4. Where and how are instance attributes created?

**Ans: Instance attributes are not shared by objects. Every object has its own copy of the instance of the instance attribute.**

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

**Ans: Self represents the instance of the class. By using the “”self” keyword we can access the attributes and methods of the class in python.**

Q6. How does a Python class handle operator overloading?

**Ans: The operator overloading in Python means provide extended meaning beyond their predefined operational meaning. Such as, we use the "+" operator for adding two integers as well as joining two strings or merging two lists. We can achieve this as the "+" operator is overloaded by the "int" class and "str" class.**

Q7. When do you consider allowing operator overloading of your classes?  
**Ans : Consider that we have two objevts which are a physical representation of a class and we have to add two objects with binary ‘+’ operator it throws an error, because compiler don’t know how to add two objects. So we define a method for an operator and process is called operator overloading.**

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

**Ans: “+”**

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

**Ans : Inheritance and polymorphimism**