**ASSIGNMENT 2**

Q1. What is the relationship between classes and modules?

Ans: A module is a python file containing many classes and methods. We can use these classes or methods by importing the module in current file.

Q2. How do you make instances and classes?

Ans: We make classes by using the syntax: class ClassName: where class is the keyword and ClassName is the class name variable. Instances are created by: instance\_name=ClassName(argument), where instance\_name is the name of the instance, ClassName is the name of class and argument is the arguments accepted by \_\_init\_\_ method in the class.

Q3. Where and how should be class attributes created?

Ans: Class attributes are created in the class. They are created by attr\_name=any\_value, where attr\_name is the name of the attribute and any\_value is the value assigned to the attribute.

Q4. Where and how are instance attributes created?

Ans: Instance attributes are created inside the \_\_init\_\_method of the class. They are the \_\_init\_\_ method arguments. For example: class ClassName:

def \_\_init\_\_(self, attr1,attr2):

self.attr1=attr1

self.attr2=attr2

In the above code attr1 and attr2 are the instance attributes of a class.

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

Ans: ‘self’ is always the first argument of any method present in class. It is a reference to the current instance of class and used to access variables accepted by class.

Q6. How does a Python class handle operator overloading?

Ans: Python class handle operator overloading by overriding the existing operator method to user-defined methods. For example, + operator has its own \_\_add\_\_ method but if we modify the \_\_add\_\_ method according to our usage we can handle overloading very effectively.

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

Ans: When we want to change conventional working of any operator to user-defined one, then we can consider operator overloading.

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

Ans: The most popular form of operator overloading is overriding existing special methods, which start and end with double underscore. For example:\_\_add\_\_() or \_\_init\_\_() methods.

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

Ans: The two most important concepts are Inheritance and polymorphism.