Python advance assignment-02

Q1. What is the relationship between classes and modules?

Classes is function where various modules are created, so modules are subset of class.

Q2. How do you make instances and classes?

Class:

```
class student(object):
    Pass
```

Instances: instances are created inside class using self keyword

```
Class student(object):
    def _init_(self,name,id)
    self.name=name
    self.id=id
```

heighlighted one are instance inside a class

Q3. Where and how should be class attributes created?

Class attributes can be created inside class directly.

Ex:

```
Class person:
  def age():
  def place():
 def salary():
```

Q4. Where and how are instance attributes created?

```
Class student(object):
    def _init_(self,name,id)
    self.name=name
    self.id=id
```

instance attribute are created using self keyword

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

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?

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?

When there is a need of other operation like string concatenation and list concatenation which are other operations then arithmetic operation we prefer overloading.

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

Most popular one is (+) Addition , however we can perform (-),(*) on strings and list elements

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

Inheritance and Polymormphism

Inheritance where child class inherit the properties of parent class.

Polymorphism where class object exhibits the multiple propertes