Q1. What is the relationship between classes and modules?

```
A module is  a collection of classes
```

Q2. How do you make instances and classes?

```
class class_name():
    pass

instance_name=class_name()
```

Q3. Where and how should be class attributes created?

```
immediately after the class deleration inside the class

class class_name():

    class_atribute1
    class_atribute2
```

Q4. Where and how are instance attributes created?

```
Inside the __init__() method of the class


class class_name():
    def __init__(self):
        self.instance_atribute1
        self.instance_atribute2
```

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

```
slef is the special keyword which is used to map the method and the atriabutes to
a specific instance
```

Q6. How does a Python class handle operator overloading?

```
Python classes uses the magic methods to handle the opreator overloading .


__add(self,-)__ for "+"
__sub__(self,-) for "-"
__mul__(self,-) for "*"
__div__(self,-) for "/"
__eq__(self,-)  for "=="
__lt__(self,-)  for "<"
__gt__(self,-)  for ">"

```

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

```
when operator overloading imporoves the rliabilty of the code then we alloe the operator overloading of classes
```

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

```
Arithmatic operator overloading is the most popular form of operator overlading
```

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

```
1. Classes and Objects
2. Encapsulation
3.Inheritence
4.Polymorphism

```