Training Report Day-7

13 June 2024

Object Oriented Programming in Python:

Object oriented programming approach allows us to club together the data and behavior so that it becomes easier to code real world scenarios.

Classes and Objects:

- Objects are real world entities. Anything you can describe in this world is an object.
- Classes on the other hand are not real. They are just a concept.

Classes are defined by using ‘class’ keyword.

```
class Mobile:
    pass
```

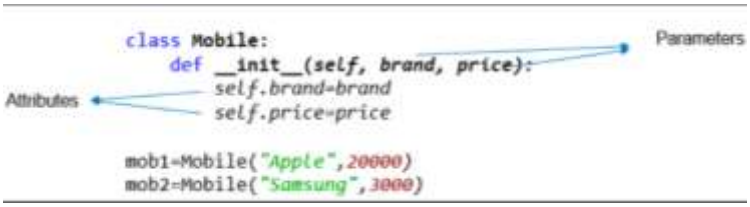
To create an object, we need a class. The syntax for creating an object is "classname()", where classname is the name of the class.

```
Mobile()
Mobile()
Mobile()
```

Attributes of an Object:

Attributes are declared by using dot(.) operator with object.

Example:



Constructor & Self Introduction:

When we create an object, the special __ init __() method inside the class of that object is invoked automatically. This special function is called as a constructor.

```
class Mobile:
    def __init__(self):
        print("Inside constructor")
mob1=Mobile()
```

self is not a keyword. self refers to the current object being executed.

```
class Mobile:
    def __init__(self):
        print("Id of self in constructor:", id(self))
mob1=Mobile()
id(mob1)
```

Parameterized constructor :

If a constructor takes parameters then it would be called as parameterized constructor.

```
class Mobile:
    def __init__(self, brand, price):
        print("Inside constructor")
        self.brand = brand
        self.price = price

mob1=Mobile("Apple", 20000)
print("Mobile 1 has brand", mob1.brand, "and price", mob1.price)

mob2=Mobile("Samsung",3000)
print("Mobile 2 has brand", mob2.brand, "and price", mob2.price)
```

```
class car:
  make = "hyundai"
  model = "verna"
  year = "2024"
obj1 = car()
print(obj1.make)
print(obj1.model)
print(obj1.year)
```

```python
# question 2
class person:
  name = ""
  age= 0
  def greet(self,a,b):
    name = a
    age = b
    print("hello",name,"you are",age,"years old")

obj2 = person()
obj2.greet("rahul",23)
```

```python
# question 3
class rectangle:
  length = 20
  breadth = 10
  def area(self):
    print(self.length*self.breadth)
obj3 = rectangle()
obj3.area()
```

```python
# question 4
class student:
  name = "lovin"
  grade = [80,80,70,90,60]
  def avg(self):
    grade = self.grade
    print(sum(grade)/len(grade))
obj4 = student()
obj4.avg()
```