EXPERIMENT - 2
OBJECTS

GROUP MEMBERS NAME:
Harshal Jain - 60004190041
Junaid Girkar - 60004190057
Khushi Chavan - 60004190061
Megh Dedhia - 60004190067

AIM: Creating objects using python.

SYNTAX:

```
<object-name> = <class-name>(<arguments>)
```

The self Parameter

The self parameter is a reference to the current instance of the class, and is used to access variables that belong to the class.
It does not have to be named self , you can call it whatever you like, but it has to be the first parameter of any function in the class:

```
class Person:
  def __init__(xyz, name, age):
    xyz.name = name
      xyz.age = age

  def myfunc(abc):
    print("Hello my name is " + abc.name)

p1 = Person("Junaid", 20)
p1.myfunc()
```

OUTPUT:

```
Hello my name is Junaid
```

Modify Object Properties

You can modify properties on objects like this:

```
p1.age = 40
```

Delete Object Properties

You can delete properties on objects or full objects itself by using the del keyword:

```
del p1.age

del p1
```

Instance Attributes

Unlike class attributes, instance attributes are not shared by objects. Every object has its own copy of the instance attribute (In case of class attributes all objects refer to a single copy).

CODE:

```
s1.name = "Mohit"
s1.age = 20
print(s1.__dict__)
```

OUTPUT:

```
{'name': 'Mohit', 'age': 20}
```

Accessing Invalid Instance Attributes will give an error.

CODE:

```
s2 = Student()
s2.rollNumber = 101

print(s2.__dict__)
print(s2.name)
```

OUTPUT:

```
AttributeError                            Traceback (most recent
call last)
<ipython-input-12-e81b27c01c56> in <module>()
----> 1 print(s2.name)
AttributeError: 'Student' object has no attribute 'name'
```

Methods:

CODE:

```
print(hasattr(s1, 'name'))
print(hasattr(s2, 'name'))
print(getattr(s1, 'name'))
print(getattr(s2, 'name', 'test'))
```

OUTPUT:

```
True
False
Mohit
test
```

CODE:

```
delattr(s1, 'name')
print(s1.__dict__)
```

OUTPUT:

```
{'age': 20}
```

Instance Method

Instance attributes are those attributes that are not shared by objects. Every object has its own copy of the instance attribute.

CODE:

```
class Student :

    ## Instance methods
    def printHello(self) :
        print("Hello")

    def print(self, str) :
        print(str)

    def printName(self) :
        print(self.name)
```

CONCLUSION: We learnt about objects and the various functionalities they have and implemented those in python