9-10-2024

Training Day – 17

OOPS: Object Oriented Programming: In python we have concept of classes and OOPS base is Class

Class: Is a collection of variables and functions. In OOPS or in class, generally the functions are called methods.

Python Supports Modular Approach Of Programming: We can develop any big project without OOPS also but complexity of the project will be increased a lot. Using OOPS, we can decrease the complexity of the project, the project is made scalable using OOPS, we can easily modify and enhance the project later using OOPS.

Syntax to create a class:

```
class Class_Name:
   variables and methods ie block of code
```

.

How to create object of any class:

```
obj_name=class_name(arguments)
obj_name=class_name()
```

```
# #New Program
# class C1:
#     pass
# obj=C1()       #obj is object of C1 Class
# print(type(obj))
# print(obj)
```

If we are calling a method which is created inside a class, then while calling from actual parameters, first object is passed to formal parameters and then rest of the arguments are passed.

"""

```
# #New Program
# d1={1:10,2:20,3:30}     #d1 is of dict type
# d2={11:100,12:200,13:300}
# res=d1.values()     #If values is a function values(d1)
# print(res)
```

When we call a method which is inside a class then from formal parameters object is passed to the first argument of actual parameters. Python recommends that the object name in formal parameters should be self.

```
# #New Program
# L1=[1,2,3]
# L1.append(5)
```

Class represents real time entity.

One of the benefits of using OOPS is that we can easily represent real time entities using classes

Syntax to call a method of a class:
obj_name.method_name(arguments)

How to access an instance variable:
obj_name.variable_name

```
# #New Program
# class C1:
#     pass
# obj1=C1()
# obj1.a=5    #a is a variable of obj1 object
# print(obj1.a)
# obj1.b=7
# obj1.c=9
# print(obj1.a,obj1.b,obj1.c)
# obj2=C1()
# obj2.a=20
# obj2.b=30
# print(obj1.a,obj1.b,obj1.c,obj2.a,obj2.b)
```