OOP.

```
class Add:
    a=1
    b=2
def __init__(self):
    self.a = 5
    self.b = 10
def calculate(self):
    print("a=", self.a)
    print("b=", self.b)
    init_sum = self(self.a + self.b)
    print("sum is", init_sum)
    print("a=", Add.a)
    print("b=", Add.b)
class_sum = Add.a + Add.b
print("Sum is", class_sum)
a1 = Add()
a1.calculate()
```

★ Classes & objects : are the two main aspects of object oriented programming. where,
class = creates a new type
objects = Are instances of the class.

Objects can store the data using ordinary variables that belong to the object.

Variables that belong to an object or class are refer to as fields.

Objects can have functionality by using the functions that belong to a class. Such functions are called methods of the class.
collectively the fields and methods are refer to as attributes of that class.

The fields are of two types :
i) They can belong to each instance or object of the class.
ii) They can belong to the class itself.
Therefore, type1 are called instance variable & type two are called class variable.

Self keyword : The self keyword in python is equivalent to "this" pointer in C++, Java & Bisha. The self keyword is a variable that can refer to the object itself by just adding it to the beginning of parameter list.

Note: If you do not give a value to this parameter when you call the method the python will provide it automatically.

The init method is run as soon as (__init__) an object of an class is created, that is passing initial value to the objects.

Be __init__ takes as the parameter name along with the self keyword.

If there are two variable the dotted notation self.name where name is the local variable that is being called.