Instruction to Object Oriented Programming in python

Instruction to Object oriented programming in Python. In Programming languages, mainly there are two approaches that are used to write program or code.

1. Procedural programming
2. Object oriented programming

The procedural we are following till now is the session, we will learn about Object oriented programming (OOP). The basic idea of Object oriented programming (OOP) in python is to use classes and objects to represent real world concepts and entities.

A class is a blueprint or template for creating objects. It defines the properties and methods that an object if that class will have. Properties are the data or state of an objects to methods are the actions it behaviors that an object can perform.

An object is an instance of a class, and it contains its own data and methods. For example, you could create a class called “Person” that has properties such as name and age, and methods such as speak () and walk (). Each instance of the Person class would be unique object with its own name and age, but they would all have the same methods to speak and walk.

One of the key features of OOP in Python is encapsulation, which means that the internal state of an object is hidden and can only be accessed or modified through the object’s methods. This helps to protect the object’s data prevent it from being modified in unexpected ways.

Another key features of OOP is inheritance, which allows new classes to be created that inherit the properties and methods of an existing class. This allows for code reuse and makes it easy to create new classes that have similar functionality to existing classes.

Polymorphism is also supported I Python, which means that objects if different classes can be treated as if they were objects of a common class. This allows for greater flexibility in code and make it easier to write that code work with multiple types of objects.

Source Code

```
# def hello():
#     print("Hello")

# hello()
# sales1=3000
# profit1=2000
# ad1=100
# #ali.sales


# sale2=4000
# profit2=3000
# ad2=200
# #kamal.sales

# sales2=5000
# profit3=4000
# ad3=300
# #ahmad.sales


# RaillwayForm ---------> Class [bluePrint]
# munawar --------> munawar ki information wala form ----> Object[entity]
# kamal---------> kamal ki info wala form -----> Object[entity]

# ali --------> ali ke information wala form  ----> Object[entity]

# #ali.changeName("Ali Ahmad")
```

Thank You