1. explain class and object with respect to object oriented programming . give a suitable example.

ANSWER =

Class is a user-defined datatype that contain its own data members and member functions. The member functions an d data members can be accessed with the help of objects. It is the primary concept of object-oriented programming. A class is used to organize information or data so a programmer can reuse the elements in multiple instances.
For example, if a programmer wants to make three instances of Dogs, Golden Retriever, poodle, and Maltese dog. T he class dog would store similar information that is unique to each dog and appropriate information will be associate d with the class Dog.

An object in OOP is a component which consists of properties to make a particular data useful. For example, let's co nsider a class Student. We can access various student details using some common property attributes like student na me, roll number, etc

2. name the four pillars of oops.

ANSWER = Inheritance, Polymorphism, Encapsulation and Abstraction.

3. explain why the __init__() function is used give a suitable example.

ANSWER = "__init__" is a reseved method in python classes. It is called as a constructor in object oriented terminol ogy. This method is called when an object is created from a class and it allows the class to initialize the attributes of t he class.

```
class Person:
    def __init__(self, name):
        self.name = name
    def say_hi(self):
        print('Hello, my name is', self.name)
p = Person('book')
p.say_hi()
```

4. why self is used in oops.

ANSWER = The self parameter is a reference to the current instance of the class, and is used to access variables that belongs 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 functio n in the class

5. what is inheritance give an example for each type of inheritance.

ANSWER = Inheritance is a mechanism in which one class acquires the property of another class. For example, a ch ild inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class. H ence, inheritance facilitates Reusability and is an important concept of OOPs

Single Inheritance:

Single inheritance enables a derived class to inherit properties from a single parent class, thus enabling code reusabil ity and the addition of new features to existing code.

Multiple Inheritance:

When a class can be derived from more than one base class this type of inheritance is called multiple inheritances.

Multilevel Inheritance :

In multilevel inheritance, features of the base class and the derived class are further inherited into the new derived cl ass. This is similar to a relationship representing a child and a grandfather.

Hierarchical Inheritance:

When more than one derived class are created from a single base this type of inheritance is called hierarchical inherit ance.

Hybrid Inheritance:

Inheritance consisting of multiple types of inheritance is called hybrid inheritance.