**1)Explain Class and Object with respect to Object-Oriented Programming. Give a suitable example.**

Ans- A class is considered as a blueprint of objects. We can think of the class as a sketch (prototype) of a house. It contains all the details about the floors, doors, windows, etc. Based on these descriptions we build the house. House is the object.

class Bike:

name = ""

gear = 0

An object is called an instance of a class.

# create class

class Bike:

name = ""

gear = 0

# create objects of class

bike1 = Bike()

**2) Name the four pillars of OOPs.**

Ans- Inheritance, Polymorphism, Encapsulation and Abstraction.

**3) Explain why the \_\_init\_\_() function is used. Give a suitable example.**

Ans- class Person:

    # init method or constructor

    def \_\_init\_\_(self, name):

        self.name = name

    # Sample Method

    def say\_hi(self):

        print('Hello, my name is', self.name)

p = Person('Nikhil')

p.say\_hi()

**4. Why self is used in OOPs?**

Ans- The self variable is used to represent the instance of the class which is often used in object-oriented programming. It works as a reference to the object. Python uses the self parameter to refer to instance attributes and methods of the class.

**5) What is inheritance? Give an example for each type of inheritance.**

There are five types of inheritances:

Single Inheritance

Multiple Inheritance

Multilevel Inheritance

Hierarchical Inheritance

Hybrid Inheritance

Single Inheritance

This type of inheritance enables a subclass or derived class to inherit properties and characteristics of the parent class, this avoids duplication of code and improves code reusability

### Multiple Inheritance

This inheritance enables a child class to inherit from more than one parent class. This type of inheritance is not supported by java classes, but python does support this kind of inheritance. It has a massive advantage if we have a requirement of gathering multiple characteristics from different classes.

### Multilevel Inheritance

In multilevel inheritance, the transfer of the properties of characteristics is done to more than one class hierarchically. To get a better visualization we can consider it as an ancestor to grandchildren relation or a root to leaf in a tree with more than one level.

### Hierarchical Inheritance

This inheritance allows a class to host as a parent class for more than one child class or subclass. This provides a benefit of sharing the functioning of methods with multiple child classes, hence avoiding code duplication.

### Hybrid Inheritance

An inheritance is said hybrid inheritance if more than one type of inheritance is implemented in the same code. This feature enables the user to utilize the feature of inheritance at its best. This satisfies the requirement of implementing a code that needs multiple inheritances in implementation.