1. **Explain all the following function in brief**

* **\_\_init\_\_(self) (i.e. constructor).**
* It is used to create constructor.
* It is a special method of the class which is used to initialize the object with properties.
* This invoke /cabll automatically when the object of class get created.
* In constructor we can define parameters also if needed.
* So there are 2 types of constructor.

1. Non-Parameterized

-It set the default value to each object.

2. Parametrized

-It set the different value to each object.

* **\_\_str\_\_(self) method**
  + It is also one of the method of object class which used convert python object into string when we print object into print method.
  + This method returns the string representation of the object.
  + This method is called when print() or str() function is invoke on an object.
* **\_\_del\_\_(self) (i.e. destructor)**
* It is used to create destructor.
* It is also special method of class which is used to destroy the object.
* And even it help to perform cleaning task on the object.
* It is also invoke automatically when the object delete from the memory.
* In python destructor are not need as much in c++ because python has Garbage Collector.

1. **Explain Inheritance in brief.**

* Inheritance is also one of the oops principle.
* In inheritance, newly created class can access properties of another class.
* The main advantage of inheritance is used to achieve reusablity of code.
* Syntax:-

class existing\_class:

….

class new\_class(existing class)

…

* Here, Existing\_class is known as parent class or also known as Super class o Base class. New class is known as child class or sub class or derived class.

1. **Explain all type of inheritance with Programs.**

In python, There are 5 types of inheritance.

**1. Single Inheritance**

* The inheritance program which has one single parent and child class is known as Single inheritance.
* E. g:

class Person:

def setdata(self):

self.fname='Raj'

self.lname='Sharma'

class Student(Person):

def showdata(self):

print("First Name:-",self.fname)

print("Last Name:-",self.lname)

**2. Multilevel Inheritance**

* In inheritance program if child class having one more child class is known as Multilevel inheritance.
* It represent as grand parent-parent-child
* E. g:

class Person:

def personinfo(self):

print('This is a Person class with Personal details')

class Employee(Person):

def salaryinfo(self):

print('This is a Employee class with Salary details')

class Manager(Employee):

def bonusinfo(self):

print("This is Manager class with Bonus details.")

**3. Hierarchical Inheritance**

* In this type one single parent class having 2 or more child classes is known hierarchical inheritance.
* E. g: Customer is person but not employee. Employee is a person but not customer.

class Person:

def personinfo(self):

print("This is a Person Class with personal details")

class Employee(Person):

def salaryinfo(self):

print("This is a Employee Class with Salary details")

class Customer(Person):

def shoppinglist(self):

print("This is a Customer Class with shopping list")

**4. Multiple Inheritance**

* In this type one single child class having 2 or more parent class.
* A single child class inherits the properties of multiple parent class.
* E. g:

class Human:

def walk(self):

print('Human can walk')

class Animal:

def jump(self):

print('Animal can Jump')

class Vampire(Human,Animal,):

pass

**5. Hybrid inheritance**

* It is combination of all type of inheritance.
* Inheritance consisting of multiple types of inheritance is called hybrid inheritance.
* E. g:

class School:

def func1(self):

print('This function is in school')

class Student1(School):

def func2(self):

print('This function is in Studen1')

class Student2(School):

def func3(self):

print('This function is in Student2')

class Student3(Student1,School):

def func4(self):

print('This function is in Student3')

obj=Student3()

obj.func1()

obj.func2()