1. **What is OOP's?**

* Oops Stands for Object Oriented Programming.
* Object Oriented Programming refers to a type of computer programming in which programmers define the data type of data structure and also the types of operations (function) that can be applied to the data structure.
* Object is a group of interrelated variable and function where variables are referred to as Properties/Attribute of object.
* And Function/Methods are referred to the behaviour of object.
* Object provides the better and clear structure for the program.
* E. g: Person is an object which has certain properties such as height, weight, gender, age etc. It also has certain methods such as move, walk, talk and so on.
* Oops is a concept followed by multiple programming language like C+, Python, Java ,C# etc

1. **What are the main features/principles of OOPs?**

The features of object oriented programming are

* **Object**
  + Object is a real world entity.
  + It is an instance of class.
  + It has a state (Properties) and behavior (functionality).
  + Syntax:-
    - Object name=Class name()
* **Class**
  + Class is a blueprint for an object. Class is a prototype for an object.
  + Class is collection of object.
  + It is also known as logical entity.
  + Syntax:-

class class name:

class body

* **Encapsulation**
  + Encapsulation is one of the oops principle, which is use to wrap a data into single unit.
  + This put restrictions on accessing variables and methods directly and can prevent the accidental modification of data.
  + It is achieved in python using private variable.
  + Private variables are those variables prefix with \_\_ and we can access them into class only.
  + To access the private variables outside a class we use getter method and setter method which are known as accessors.
* **Inheritance**
  + Inheritance is a mechanism in which one class acquires the property of another class.
  + The main advantage of inheritance is used to achieve Reusability of code.
  + For example, a child inherits the traits of his/her parents.
  + Syntax:-

class Parent\_class:

……..

class child\_class (Parent\_class):

……..

* There are 5 types of inheritance i. e single, multilevel, hierarchical, multiple, hybrid.
* **Polymorphism**
  + In literal, Polymorphism means the ability to take various forms.
  + In python, Polymorphism allows us to define methods in the child class with same name as defined in their parent class.
  + It refers to use of single property or method differently in different scenarios.
  + We can achieve polymorphism in python using method overriding.
  + Parent and child class having method with same name is known as Method Overriding.
* **Abstraction**
  + Abstraction in python is achieved by using abstract classes and interfaces.
  + An abstract class is a class that generally provides incomplete functionality and contains one or more abstract methods.
  + It is used to avoid the complexity of application by hiding irrelevant details.
  + In python, Abstract class must be inherit to ABC class of abc module.
  + Syntax:-

From abc import ABC

class class name(Abc):

body of class

* Abstract method is declared with @abstractmethod decorator of abc module.

1. **What is Class and explain with example?**
   * Class is a blueprint for an object. Class is a prototype for an object.
   * Class is collection of object.
   * It is also known as logical entity.
   * Syntax:-

class class name:

class body

* E. g:-

Class Student:

Pass

It is an empty class. Pass is a keyword to define any block of code.

1. **What is Object explain with example?**
   * Object is a real world entity.
   * It is an instance of class.
   * It has a state (Properties) and behavior (functionality).
   * Syntax:-

Object name=Class name()

* E. g:-

Class Student:

def prop(self):

self.id=’101’

self.name=’Roshan’

def show(self):

print(‘Id:-’,self.id)

print(‘Name:-’,self.name)

std=Student()

std.prop()

std.show()

* In above example, one class is created named Student having its properties and methods.
* std is a object of Student class. Using that object we can called the method and properties of class.

1. **What is Constructor and Destructor and its properties?**
   * 1. **Constructor:-**

* Constructor is method of class which is used to initialize the object with properties.
* In python name of constructor of each class is same i. e **def \_\_init\_\_(self):**
* This constructor invokes automatically when the object of class get created.
* Constructor cannot return values as it always return object of that class.
* There are two types of constructor
  1. **Parameterized Constructor**
* It set the default value to each object.
* E. g:

class student:

def \_\_init\_\_(self):

self.id=101

self.name=’Raj’

def show(self):

print(f'Hello {self.name} your id is {self.id}')

std=student()

std.show()

* 1. **Non-Parameterized Constructor**
* It set different value to each object.
* E. g:

class student:

def \_\_init\_\_(self,i,n):

self.id=i

self.name=n

def show(self):

print(f'Hello {self.name} your id is {self.id}')

std=student(101,'Roshan')

std.show()

* + 1. **Destructor:-**
* Destructor is also special method of class which is used to destroy the object.
* And help to perform cleaning task on object.
* It will execute automatically whenever process of object finished.
* In python destructor are not needed as much in c++ because python has garbage collector.
* **def \_\_del\_\_(self):** is used to create destructor.