**DPP-06 Theory Topic**

**OPPs Easy Questions**

1. What is a class in Python?

A class in Python is a blueprint or template for creating objects and is used to group things together. A class is created using the “class” keyword.

Ex:- class mine:

2. How do you define a class in Python?

We can define a class in python by using the “class” keyword.

Ex:-

class mine:

Var = 10

def display():

print(mine.Var)

3. What is an object in Python?

Objects in python are instances of a class. Objects in python are used to access the functionalities of the class. An object is the copy of the class with the actual values. Object consists of state behaviour and identity

4. How do you create an object from a class?

We can create an object of the class using class name.

Ex:-

Class myclass:

Name = ”myname”

M1 = myclass()

#M1 is the object which going to used for accessing the methods and attributes.

5. What is the \_\_init\_\_ method?

The python “\_\_init\_\_” method is declared within a class and is used to initialize the attributes of an object as soon as the object is formed. While defining the \_\_init\_\_() method, a default parameter, named 'self' is always passed in its argument. This self represents the object of the class itself.

Example:

# A Sample class with init method

class Person:

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

self.name = name

def display(self):

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

p = Person('Alpha')

p.display()

6. What is a class attribute?

A class attribute is a python variable which is associated with the class rather than the particular object. It can be any value which is used.

Ex:-

class example\_lass:

count = 0 # class attribute

cl = example\_class()

print(cl.count)

7. What is an instance attribute?

The variable which is defined within the class constructor method (\_\_init\_\_) in python is known as the instance attribute. The instance attribute of the class is unique for each object of the class. The instance are accessed through self keyword.

**Ex:-**

class Car:

def \_\_init\_\_(self, make, model):

self.make = make

self.model = model

car1 = Car("Toyota", "Corolla")

car2 = Car("Honda", "Civic")

print(car1.make)

print(car2.make)

**output**:-

Toyota

Honda

8. How do you access an attribute of an object?

9. What is a method in a class?

○ Describe what a method is and how it differs from a function.

10. How do you define a method within a class?

○ Write a simple example of a class with a method.

11. What does the self keyword represent?

○ Explain the role of the self keyword in class methods.

12. Can you create a class without any attributes or methods?

○ Discuss whether it’s possible and provide a brief code example.

**Mid-Level Questions**

13. How can you create a class that inherits from another class?

○ Provide an example of inheritance in Python.

14. What is polymorphism in the context of classes?

○ Explain polymorphism and give a code example.

15. What is encapsulation, and how is it achieved in Python?

○ Describe encapsulation and provide an example of private attributes.

16. How do you create a class method in Python?

○ Explain the concept of a class method and give an example.

17. What is the difference between a class method and a static method?

○ Compare class methods and static methods with code examples.

18. How can you override a method in a subclass?

○ Write an example showing method overriding in inheritance.

19. What are magic methods in Python?

○ Define magic methods and provide examples of commonly used ones (e.g., \_\_str\_\_, \_\_repr\_\_).

20. How can you create a property in a class?

○ Explain how to use the property decorator to create a property and provide an example.