

## ASSIGNMENT-5

Q1, Create a vehicle class with an init method having instance variables as name\_of\_vehicle, max\_speed

and average\_of\_vehicle.

```
[64]: class vehicle:
      def __init__(old,name_of_vehicle,max_speed,avg_of_vehicle):
          old.name_of_vehicle1=name_of_vehicle
          old.max_speed1=max_speed
          old.avg_of_vehicle1=avg_of_vehicle
      def return_vehicle_details(old):
          return old.name_of_vehicle1, old.max_speed1, old.avg_of_vehicle1

[65]: vehicles=vehicle("honda",200,3000)

[69]: vehicles.return_vehicle_details()

[69]: ('honda', 200, 3000)
```

Q2. Create a child class car from the vehicle class created in Que 1, which will inherit the vehicle class. Create a method named seating\_capacity which takes capacity as an argument and returns the name of the vehicle and its seating capacity.

```
[16]: class Vehicle:
      def __init__(self, name, max_speed):
          self.name = name
          self.max_speed = max_speed

      class Bus(Vehicle):
          pass

      School_bus = Bus("School Volvo", 180)
      print("Vehicle Name:", School_bus.name, "Speed:", School_bus.max_speed)

      class bus(Vehicle):
          def seating_capacity(self, capacity=50):
              return f"The seating capacity of a {self.name} is {capacity} passengers"

      bus("school bus", 120).seating_capacity()

      Vehicle Name: School Volvo Speed: 180

[16]: 'The seating capacity of a school bus is 50 passengers'
```

Q3. What is multiple inheritance? Write a python code to demonstrate multiple inheritance.

**A-** A class can be derived from more than one superclass in Python. This is called multiple inheritance.

```
0]: class Animals:
    def animi_info(self):
        print("This is about tiger")

    class Fishes:
        def fish_info(self):
            print("This is about fishes")

    class All(Animals ,Fishes):s
        pass

2]: every_animals= All()

every_animals.animi_info()
every_animals.fish_info()

This is about tiger
This is about fishes
```

Q4. What are getter and setter in python? Create a class and create a getter and a setter method in this class.

**A-** A class can have one more variables (sometimes called properties). When you create objects each of those objects have unique values for those variables.

Class variables need not be set directly: they can be set using class methods. This is the object orientated way and helps you avoid mistakes.

```

]: class Friend:
    def __init__(self):
        self.job = "None"

    def getJob(self):
        return self.job

    def setJob(self, job):
        self.job = job

Alice = Friend()
Bob = Friend()

Alice.setJob("Carpenter")
Bob.setJob("Builder")

print(Bob.job)
print(Alice.job)

Builder
Carpenter

```

Q5.What is method overriding in python? Write a python code to demonstrate method overriding.

**A-** Method overriding is a concept in object-oriented programming where a subclass provides a specific implementation of a method that is already defined in its parent class. This allows the subclass to customize or extend the behavior of the inherited method while keeping the method name and signature the same. Method overriding is a way to implement polymorphism in Python.

```
]: class Animal:
    def speak(self):
        print("Animal speaks")

class Dog(Animal):
    def speak(self):
        print("Dog barks")

class Cat(Animal):
    def speak(self):
        print("Cat meows")

# Creating instances of Dog and Cat
dog = Dog()
cat = Cat()

# Calling the speak method on instances
dog.speak()
cat.speak()
```

```
Dog barks
Cat meows
```