POINTMENT/MEETING

9 feb Assignment

1. create a child class car from the vehicle class created in Ques 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 & its seating capacity.

```
childclass Vehicle:
    def __init__( name of vehicle, capacity):
         Self.name of vehicle = name of vehicle
        Self.capacity =capacity
    child class car(vehicle):
        pass
child class_car = car("School car", 8)
  print (isinstance(childclass, Vehicle))
      True.
```

2. What is multiple inheritance? write a python code to demonstrate multiple inheritance.

when a class is derived from more than one base class it is called multiple inheritance.

TMENT NOTES

```
   Class Base 1: Body of the class
  class Bas 2: Body of the class
  class derived(Base1, Base 2): Body of the class
```

Q.3 Create a vehicle class with an init method having instance variables as name_of_vehicle, max_speed & average_of_vehicle.

```
class Vehicle:
def __init__(Self, name, max_speed, mileage):
        Self.name = name
       self.nam=speed = max_speed
       self.mileage = mileagl
```

Q. 4 What are getter & setter in python? Create a class & create a getter & a setter method in this class?

→ In Python, getters and setters are not the same as those in other object oriented programming languages. Basically, the main purpose of using getters & setters in object-oriented programs is to ensure data encapsulation.

```
class Geeks:
    def __init__(self):
        self._age = 0

def get_age(self):
    print("getter method called")
    return self._age
```

```
def set_age(self,a):
print("setter method called")
    self._age =a

def del_age(self):
    del self._age

 age =property(get_age,set_age,del_age)
 mark =Geeks()

 mark.age =10
 print(mark.age)
```

Q.5 What is Method overiding in python? write a python code to ~~programming~~ demonstrate method overriding.

→ Method overriding is an ability of any object-oriented programming language that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super classes or parent classes.

→ When a method in a subclass has the same name, same parameters or signature & same return type (or sub-type) as a method in its super class, then the Method in the subclass is said to override the Method in the super class.

APPOINTMENT NOTES

12 JULY
THURSDAY

193-172 • WK 28

JUL													
M	T	W	T	F	S	S	M	T	W	T	F	S	S
						1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31					

APPOINTMENT/MEETING

9

10

11

12

1

2

Ex:-
```
class parent ()
    def __init__(self):
        self.value = "Inside Parent"
    def show(self):
        print(self.value)

class child (Parent):
    def __init__(self):
        self.value = "Inside child"
    def show(self):
        print(self.value)
```