Assignment 10: 09 Feb 2023

Q1, Create a vehicle class with an init method having instance variables as name_of_vehicle, max_speed and average of vehicle.
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.
Q3. What is multiple inheritance? Write a python code to demonstrate multiple inheritance.
Q4. What are getter and setter in python? Create a class and create a getter and a setter method in this class.
Q5. What is method overriding in python? Write a python code to demonstrate method overriding.

Q1, Create a vehicle class with an init method having instance variables as name_of_vehicle, max_speed and average of vehicle.

Ans:

```
class vehicle:
    def __init__ (self,name_of_vehicle, max_speed, average_of_vehicle ):
        self.name_of_vehicle=name_of_vehicle
        self.max_speed=max_speed
        self.average_of_vehicle=average_of_vehicle
```

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.

Ans:

```
class car(vehicle):
    def __init__ (self,name_of_vehicle, max_speed, average_of_vehicle, capacity ):
        super().__init__(name_of_vehicle, max_speed, average_of_vehicle)
        self.capacity=capacity
    def show_name_capacity(self):
        return self.name_of_vehicle, self.capacity
c1=car("Honda 120", 180, 60, 4)
c1.show_name_capacity()
```

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

Ans: When two different classes combined to create another class than the created new class is called multiple inheritance

```
class Mom: # Create mom
    def eye_color(self):
        return ("Eye Color Is Blue : I am Mom Eye")
class Dad: # create dad
    def skin_color(self):
        return "Skin Colour Is white : I am Dad skin"
class Baby(Mom, Dad): # create baby
    pass
baby1= Baby()#create baby object
```

```
print(baby1.eye_color())# see mom gene
print(baby1.skin_color()) # see dad gene
```

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

Ans:

Getter: A method that allows you to access an attribute in a given class

Setter: A method that allows you to set or mutate the value of an attribute in a class

```
class Getter_Setter:
    def __init__(self, age = 0):
         self._age = age
      # using the getter method
    def get_age(self):
        return self._age
      # using the setter method
    def set_age(self, a):
        self._age = a
Arun = Getter_Setter()
#using the setter function
Arun.set_age(27)
# using the getter function
print(Arun.get_age())
print(Arun._age)
```

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

Ans: The method overriding in Python means creating two methods with the same name but differ in the programming logic. The concept of Method overriding allows us to change or override the Parent Class function in the Child Class.

```
# Defining parent class
class Parent1():
    # Constructor
    def __init__(self):
        self.value = "Inside Parent"
    # Parent's show method
    def show(self):
        print(self.value)
# Defining child class
class Child(Parent1):
    # Constructor
    def __init__(self):
        self.value = "Inside Child"
    # Child's show method
    def show(self):
        print(self.value)
# Driver's code
obj1 = Parent()
```

```
obj2 = Child()
obj1.show()
obj2.show()
```