

Assignment8 oops concept

March 7, 2023

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

```
[28]: class vehicle:

    def __init__(self,name_of_vehicle,max_speed,avg_vehicle):
        self.name_of_vehicle=name_of_vehicle
        self.max_speed=max_speed
        self.avg_vehicle=avg_vehicle

    def return_vehicles_details(self):
        return self.name_of_vehicle,self.max_speed,self.avg_vehicle
```

```
[29]: vehicle_details=vehicle('toyato',120,50)
```

```
[31]: vehicle_details.return_vehicles_details()
```

```
[31]: ('toyato', 120, 50)
```

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.

```
[38]: class car(vehicle):

    def seating_capacity(self,capacity):
        self.capacity=capacity
        self.vehicle_name=vehicle_name
```

```
[41]: def return_vehicles_details1(self):
        return self.name_of_vehicle,self.capacity
```

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

ans:-multiple class inherit in single class

```
[1]: class class1():
      def class_1(self):
          return 'this is my first class'
```

```
[3]: class class2():
      def class_2(self):
          return 'this is my second class'
```

```
[8]: class class3(class1,class2):
      pass
```

```
[9]: object_class=class3()
```

```
[10]: object_class.class_1()
```

```
[10]: 'this is my first class'
```

```
[11]: object_class.class_2()
```

```
[11]: 'this is my second class'
```

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 given class. setter:- a method that allows you to set or mutate the value of an attribute in a class.

```
[12]: class Geek:
      def __init__(self, age = 0):
          self._age = age

      # getter method
      def get_age(self):
          return self._age

      # setter method
      def set_age(self, x):
          self._age = x
```

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

ans:-when you have two methods with the same name that each perform different tasks.

```
[13]: class college:
      def info(self):
          print('Hello, I am the parent class')

      class student(college):
```

```
def info(self):  
    print('Hello, I am the child class')
```

```
[ ]:
```