OBJECT ORIENTED PROGRAMMING EXERCISE

OOP Exercise 1: Create a Vehicle class with max_speed and mileage As Data Members

CODE:

```
class vehicle:
  def __init__(self,max_speed,mileage):
   x=max_speed
   y=mileage
print(type(vehicle))


output: <class 'type'>
```

OOP Exercise 2: Create a Vehicle class without any variables and method.

CODE:

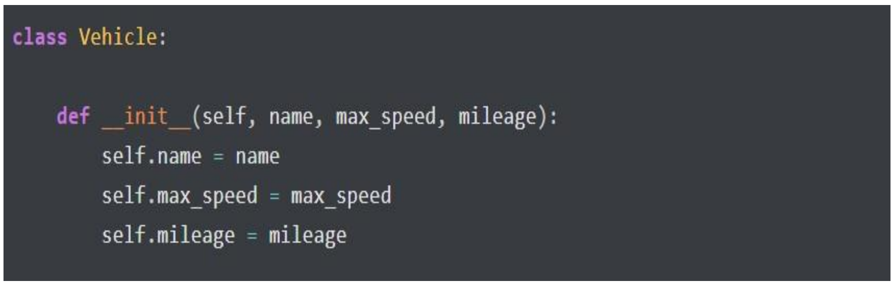
```
class vehicle:
  pass
  print(type(vehicle))
```

OUTPUT:

```
<class 'type'>
```

OOP Exercise 3: Create a child class Bus that will inherit all of the variables and methods of the Vehicle class

Given:



```
class Vehicle:

    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage
```

CODE:

```
class vehicle:
  def __init__(self,max_speed,mileage):
    self.max_speed=max_speed
    self.mileage=mileage
  def displayfunc(self):
    print("The max. speed of vehicle is: ",self.max_speed)
    print("The mileage of vehicle is: ",self.mileage)
```

```
class Bus(vehicle):
  def __init__(self,max_speed,mileage):
    super().__init__(max_speed,mileage)
volvo=Bus(130,20)
print(volvo.displayfunc()
```

OUTPUT:

```
The max. speed of vehicle is:  130
The mileage of vehicle is:  20
None
```

OOP Exercise 4: Define property that should have the same value for every class instance

Define a **class** attribute"**color**" with a default value **white**. I.e., Every Vehicle should be white.

Use the following code for this exercise.

CODE:

```
class Vehicle:
    color = "White"
    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

class Bus(Vehicle):
    pass

School_bus = Bus("School Volvo", 12, 50)
print("the color of vehicle is: ",School_bus.color,",","bus name is
: ", School_bus.name,",", "Speed:", School_bus.max_speed,"," ,"Mile
age:", School_bus.mileage)
```

OUTPUT:

```
the color of vehicle is:  White , bus name is:  School Volvo ,
Speed: 12 , Mileage: 50
```

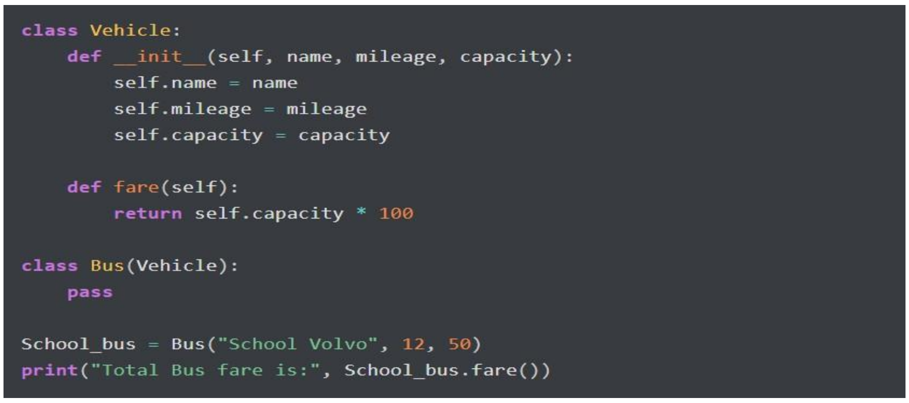
OOP Exercise 5: Class Inheritance

Given:

Create a **Bus** child class that inherits from the Vehicle class. The default fare charge of any vehicle is **seating capacity * 100**. If Vehicle is **Bus** instance, we need to add an extra 10% on full fare as a maintenance charge. So total fare for bus instance will become the **final amount = total fare + 10% of the total fare.**

Note: The bus seating capacity is **50**. so the final fare amount should be **5500.** You need to override the `fare()` method of a Vehicle class in Bus class.

Use the following code for your parent Vehicle class. We need to access the parent class from inside a method of a child class.


```python
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

    def fare(self):
        return self.capacity * 100

class Bus(Vehicle):
    pass

School_bus = Bus("School Volvo", 12, 50)
print("Total Bus fare is:", School_bus.fare())
```

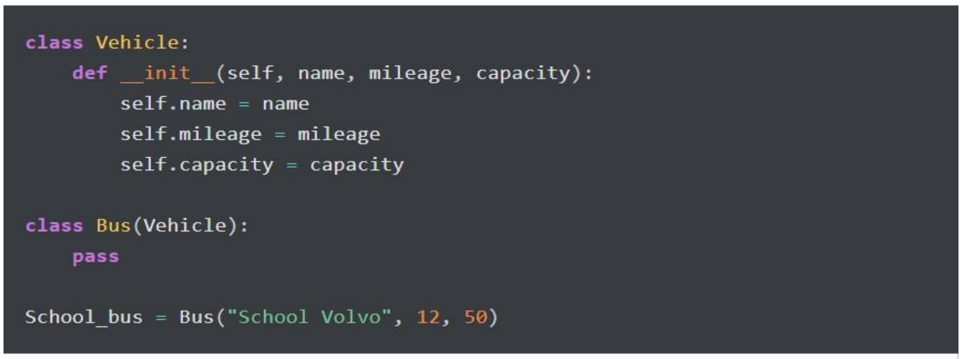
CODE:

```python
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity
    def fare(self):
        return self.capacity * 100

class Bus(Vehicle):
  pass
  def fare(self):
    a = super().fare()
    a += a * 10 / 100
    return a

School_bus = Bus("School Volvo", 12, 50)
print("Total Bus fare is:",School_bus.fare(),"Rs.")
```

OUTPUT:

```
Total Bus fare is: 5500.0 Rs.
```

OOP Exercise 6: Determine which class a given Bus object belongs to (Check type of an object)


```python
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

class Bus(Vehicle):
    pass

School_bus = Bus("School Volvo", 12, 50)
```

CODE:

```
class Vehicle:
    def __init__(self, name, mileage, capacity):
        self.name = name
        self.mileage = mileage
        self.capacity = capacity

class Bus(Vehicle):
    pass

School_bus = Bus("School Volvo", 12, 50)
print(type(School_bus))
```

OUTPUT:

```
<class '__main__.Bus'>
```