**Assignment for today**

**OOP Exercise 1: Create a Class with instance attributes**

Write a Python program to create a Vehicle class with max\_speed and mileage instance attributes.

class Vehicle:  
 def \_\_init\_\_(self, max\_speed, mileage):  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
modelX = Vehicle(240, 18)  
print(modelX.max\_speed, modelX.mileage)

### OOP Exercise 2: Create a Vehicle class without any variables and methods

class Vehicle:  
 pass

**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

Create a Bus object that will inherit all of the variables and methods of the parent Vehicle class and display it.

**Expected Output:**

Vehicle Name: School Volvo Speed: 180 Mileage: 12

class Vehicle:  
  
 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", 180, 12)  
print("Vehicle Name:", School\_bus.name, "Speed:", School\_bus.max\_speed, "Mileage:", School\_bus.mileage)

**OOP Exercise 4: Class Inheritance**

**Given**:

Create a **Bus** class that inherits from the **Vehicle** class. Give the capacity argument of Bus.seating\_capacity() a **default** value of 50.

Use the following code for your parent Vehicle class.

class Vehicle:

    def \_\_init\_\_(self, name, max\_speed, mileage):

        self.name = name

        self.max\_speed = max\_speed

        self.mileage = mileage

    def seating\_capacity(self, capacity):

        return f"The seating capacity of a {self.name} is {capacity} passengers"

**Expected Output**:

The seating capacity of a bus is 50 passengers

class Vehicle:  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
 def seating\_capacity(self, capacity):  
 return f"The seating capacity of a {self.name} is {capacity} passengers"  
  
class Bus(Vehicle):  
 # assign default value to capacity  
 def seating\_capacity(self, capacity=50):  
 return super().seating\_capacity(capacity=50)  
  
School\_bus = Bus("School Volvo", 180, 12)  
print(School\_bus.seating\_capacity())

**OOP Exercise 5: Define a property that must have the same value for every class instance (object)**

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

Use the following code for this exercise.

class Vehicle:

    def \_\_init\_\_(self, name, max\_speed, mileage):

        self.name = name

        self.max\_speed = max\_speed

        self.mileage = mileage

class Bus(Vehicle):

    pass

class Car(Vehicle):

    pass

**Expected Output**:

Color: White, Vehicle name: School Volvo, Speed: 180, Mileage: 12

Color: White, Vehicle name: Audi Q5, Speed: 240, Mileage: 18

class Vehicle:  
 # Class attribute  
 color = "White"  
  
 def \_\_init\_\_(self, name, max\_speed, mileage):  
 self.name = name  
 self.max\_speed = max\_speed  
 self.mileage = mileage  
  
class Bus(Vehicle):  
 pass  
  
class Car(Vehicle):  
 pass  
  
School\_bus = Bus("School Volvo", 180, 12)  
print(School\_bus.color, School\_bus.name, "Speed:", School\_bus.max\_speed, "Mileage:", School\_bus.mileage)  
  
car = Car("Audi Q5", 240, 18)  
print(car.color, car.name, "Speed:", car.max\_speed, "Mileage:", car.mileage)

**OOP Exercise 6: 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.

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())

**Expected Output**:

Total Bus fare is: 5500.0

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):  
 def fare(self):  
 amount = super().fare()  
 amount += amount \* 10 / 100  
 return amount  
  
School\_bus = Bus("School Volvo", 12, 50)  
print("Total Bus fare is:", School\_bus.fare())

**OOP Exercise 7: Check type of an object**

Write a program to determine which class a given Bus object belongs to.

**Given**:

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)

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)  
  
# Python's built-in type()  
print(type(School\_bus))

**OOP Exercise 8: Determine if School\_bus is also an instance of the Vehicle class**

**Given**:

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)

Assignment 2 Create this landing page

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)  
  
# Python's built-in isinstance() function  
print(isinstance(School\_bus, Vehicle))