

|  |  |
| --- | --- |
| Name | Ali Mustafa Shah |
| Roll Number | SU92-BSDSM-S24-005 |
| Section | 2A |
| Semester | 2nd |
| Subject | Opp lab |
| Task | 7 |
| Submitted by | Syed Ali Mustafa Shah |
| Submitted to | Sir Rasikh |

**Question 1**

**Task 1; Create a Python program to model a vehicle rental system:**

**1. Implement a parent class called Vehicle with attributes make and model. Include a method display\_info to display the make and model of the vehicle.**

**2. Create a child class Car inheriting from Vehicle. The Car class should have an additional attribute num\_doors and a method additional\_info to display the number of doors.**

**3. Create another child class LuxuryCar inheriting from Car. The LuxuryCar class should have an additional attribute features and a method additional\_info to display the luxury features.**

**Task 2; Create a Python program to model a company's employee hierarchy:**

**1. Implement a parent class called Employee with attributes name and position. Include a method display\_info to display the employee's name and position.**

**2. Create two child classes Manager and Worker inheriting from Employee. Each should have an additional attribute (department for Manager and hours\_worked for Worker) and a method additional\_info to display the additional attributes**

**Solution**

class Vehicle:

    def \_\_init\_\_(self, make, model):

*self*.make = make

*self*.model = model

    def display\_info(self):

        print(f"Vehicle Make: {*self*.make}, Model: {*self*.model}")

class Car(Vehicle):

    def \_\_init\_\_(self, make, model, num\_doors):

        super().\_\_init\_\_(make, model)

*self*.num\_doors = num\_doors

    def additional\_info(self):

        print(f"Number of Doors: {*self*.num\_doors}")

class LuxuryCar(Car):

    def \_\_init\_\_(self, make, model, num\_doors, features):

        super().\_\_init\_\_(make, model, num\_doors)

*self*.features = features

    def additional\_info(self):

        super().additional\_info()

        print(f"Luxury Features: {*self*.features}")

luxury\_car = LuxuryCar("BMW", "X7", 4, ["black seets", "OOP LAB", "kuch be"])

luxury\_car.display\_info()

luxury\_car.additional\_info()