BAHRIA UNIVERSITY (KARACHI CAMPUS)

Object Oriented Programming (CSC- 210)

Assignment 01

Spring 2023

Class: BSE 2B | **Shift: Morning**
Course Instructor: Engr. MAHAWISH | **Due Date: 24 Mar 2023**
Assignment Date: 12 Mar 2023 | **Marks: 05 Points**
Student Name: ABDULLAH | **Registration #: 81962**

Question: Develop an application using Object Oriented Programming concept. The application should have the following necessary elements.

i. The application should use Object Oriented Programming concepts.
ii. The application should have at least three classes with their respective attributes and methods.
iii. A proper description of the application's tasks, and the behavior and characteristics of its objects should be provided.
iv. A UML diagram should be designed to show each class and its parent-child relationships, if applicable.
v. A Python code should be written to implement the application.

Solution:

Car Rental and Sale Management System

The application's code defines four classes: Car, Rent, Sale, and Admin.

The Car class is the base class for all cars. It has an __init__ method that takes six parameters: __made, __cname, __model, __transmission, __steering, and __price. These parameters are used to initialize instance variables that represent the make, car name, model, transmission, steering type, and price per day of the car. The class also has an Info method that prints out the details of the car.

The Rent class is a subclass of the Car class. It has an additional class variable time which represents the number of days the car is rented for, and instance variables name, cont, and address that represent the name, contact number, and address of the customer. The Rent class has a method called Charges that calculates the rental charges for the car based on the number of days it is rented for and the price per day of the car. The class also has a Customer method that takes input from the customer for their details and the number of days they want to rent the car for. The Info method is overridden in this class to include the customer details and rental charges.

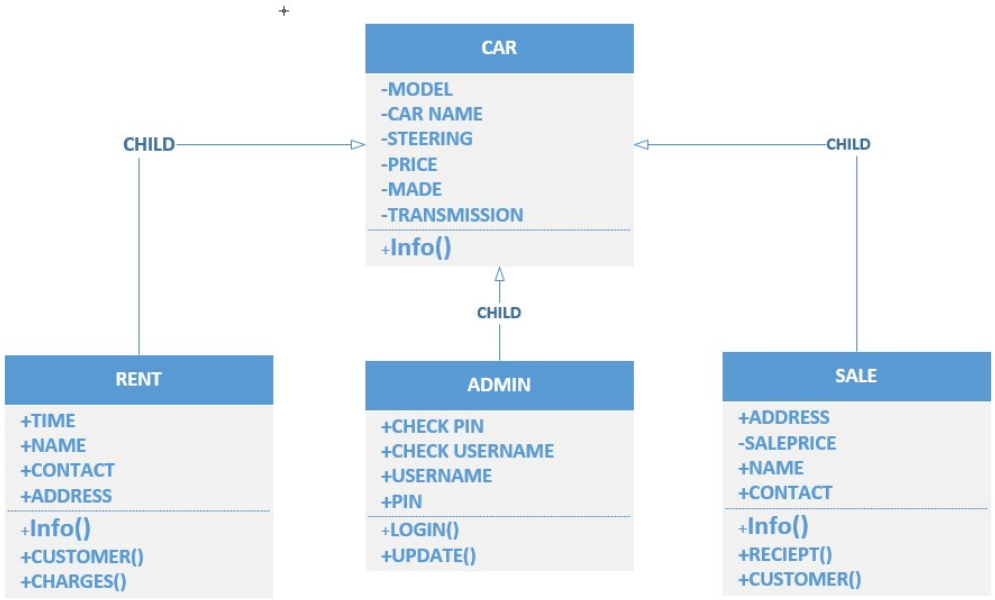
The Sale class is also a subclass of the Car class. It has an additional instance variable saleprice that represents the sale price of the car. The class also has name, cont, and address instance variables that represent the name, contact number, and address of the customer. The Sale class has a customer method that takes input from the customer for their details. The Info method is overridden in this class to include the sale price of the car, and the class has a Reciept method that prints out the customer details and sale price of the car.

The admin class is a subclass of the Car class. It has class variables username and pin that represent the login credentials for the administrator. The class has instance variables checkusername and checkpin that are used to check the administrator's login credentials. The Admin class has a Login method that prompts the administrator to enter their login credentials and checks if they are valid. If the login credentials are valid, the administrator is logged in. The class also has an Update method that prompts the administrator to select what changes they want to make to a particular car's details and makes the changes accordingly.

The code also creates four Car objects p, p1, p2, and p3. These objects represent four different cars, each with their own details such as make, car name, model, transmission, steering type, and price per day. The code also creates an Admin object a, which is used to log in the administrator and make changes to the car details.

Overall, the code is a simple implementation of a car rental and sale system, with different classes representing different aspects of the system.

UML Diagram:



Code:

```
class Car:

    def __init__(self, __made, __cname, __model, __transmission, __steering,
__price):
        self.model = __model
        self.cname = __cname
        self.made = __made
        self.transmission = __transmission
        self.steering = __steering
        self.price = __price

    def Info(self):
        print(f"MADE           : {self.made}")
        print(f"CAR NAME       : {self.cname}")
        print(f"MODEL          : {self.model}")
        print(f"TRANSMISSION   : {self.transmission}")
        print(f"STEERING TYPE  : {self.steering}")
        print(f"PRICE PER DAY  : Rs{self.price}")

class Rent(Car):

    time = 0
    name = ""
    cont = 0
    address = ""

    def Charges(self):
        return self.price*self.time

    def Customer(self):
        self.name = input("\nEnter Your Name : ")
        self.cont = int(input("Enter Contact.No : "))
        self.address = input("Enter Address : ")
        self.time = int(
            input("For how many days do you want to rent a car : "))

    def Info(self):
        print(f"CUSTOMER NAME   : {self.name}")
        print(f"CONTACT NUMBER  : {self.cont}")
        print(f"ADDRESS         : {self.address}")
        print(f"MADE            : {self.made}")
        print(f"CAR NAME        : {self.cname}")
        print(f"MODEL           : {self.model}")
        print(f"TRANSMISSION    : {self.transmission}")
        print(f"STEERING TYPE   : {self.steering}")
        print(f"PRICE PER DAY   : {self.price}")
        print(f"CAR RENTED FOR  : {self.time} DAYS")
        print(f"TOTAL CHARGES   : Rs {self.Charges()}")


class Sale(Car):
```

```
    def __init__(self, __made, __cname, __model, __transmission, __steering, __price,
__saleprice):
        super().__init__(__made, __cname, __model, __transmission, __steering,
__price)
        self.saleprice = __saleprice
        self.name = ""
        self.cont = 0
        self.address = ""

    def Customer(self):
        self.name = input("\nEnter Your Name : ")
        self.cont = int(input("Enter Contact.No : "))
        self.address = input("Enter Address : ")

    def Info(self):
        print(f"MADE           : {self.made}")
        print(f"CAR NAME       : {self.cname}")
        print(f"MODEL          : {self.model}")
        print(f"TRANSMISSION   : {self.transmission}")
        print(f"STEERING TYPE  : {self.steering}")
        print(f"SALE PRICE     : Rs{self.saleprice}")

    def Reciept(self):
        print(f"CUSTOMER NAME   : {self.name}")
        print(f"CONTACT NUMBER  : {self.cont}")
        print(f"ADDRESS         : {self.address}")
        print(f"MADE            : {self.made}")
        print(f"CAR NAME        : {self.cname}")
        print(f"MODEL           : {self.model}")
        print(f"TRANSMISSION    : {self.transmission}")
        print(f"STEERING TYPE   : {self.steering}")
        print(f"SALE PRICE      : Rs{self.saleprice}")

class Admin(Car):

    username = "admin"
    pin = 12345
    checkusername = ""
    checkpin = 0

    def Login(self):
        print("\n\n~~~~~Login Page~~~~~\n\n")
        while(self.checkusername != self.username):
            while(self.checkpin != self.pin):
                self.checkusername = input("Enter username : ")
                if(self.checkusername == self.username):
                    self.checkpin = int(input("Enter pin : "))
                    if(self.checkpin == self.pin):
                        print("\nLogged In Successfully\n")
                    else:
                        print("\nInvalid Pin")
                else:
                    print("\ninvalid Username\n")

    def Update(self):
```

```
        print("\nWhat changes do you want to make for this Rent A Car (Select from
following list)\n")
        print("1)MADE")
        print("2)CAR NAME")
        print("3)MODEL")
        print("4)TRANSMISSION")
        print("5)STEERING TYPE")
        print("6)PRICE PER DAY")
        num = int(input("\nType Option Number : "))
        match(num):
            case 1:
                self.made = input("\nMade : ")
                self.Info()

            case 2:
                self.cname = input(f"\nCar Name : ")
                self.Info()

            case 3:
                self.model = input("\nModel : ")
                self.Info()

            case 4:
                self.transmission = input("\nTransmission : ")
                self.Info()

            case 5:
                self.steering = input(f"\nSteering Type : ")
                self.Info()

            case 6:
                self.price = input("\nPrice/Day : ")
                self.Info()

            case _:
                print("\ninvalid option")

p = Car("TOYOTA", "GRANDE", "2020", "MANUAL", "Power", 15000)
p1 = Car("NISSAN", "DAYZ", "2018", "Auto", "Power", 11000)
p2 = Car("SUZUKI", "WAGON-R", "2015", "Auto", "Power", 10000)
p3 = Car("HONDA", "CIVIC", "2022", "Auto", "Power", 20000)

a = Admin("TOYOTA", "GRANDE", "2020", "MANUAL", "Power", 15000)
a1 = Admin("NISSAN", "DAYZ", "2018", "Auto", "Power", 110000)
a2 = Admin("SUZUKI", "WAGON-R", "2015", "Auto", "Power", 100000)
a3 = Admin("HONDA", "CIVIC", "2022", "Auto", "Power", 20000)

s = Sale("TOYOTA", "GRANDE", "2020", "MANUAL", "Power", 0, 2500000)
s1 = Sale("NISSAN", "DAYZ", "2018", "Auto", "Power", 0, 1800000)
s2 = Sale("SUZUKI", "WAGON-R", "2015", "Auto", "Power", 0, 1700000)
s3 = Sale("HONDA", "CIVIC", "2022", "Auto", "Power", 0, 8000000)

print("Select Option\n\n1)Admin\n2)User")
ser = int(input("\nType Here : "))
```

```
if(ser == 1):
    a.Login()
    print("\tCAR 1\n")
    p.Info()

    print("\n\tCAR 2\n")
    p1.Info()

    print("\n\tCAR 3\n")
    p2.Info()

    print("\n\tCAR 4\n")
    p3.Info()
    change = int(
        input("\n\nIn which car list do you want to make changes (Type Number) : "))
    match(change):
        case 1:
            a.Update()
        case 2:
            a1.Update()
        case 3:
            a2.Update()
        case 4:
            a3.Update()
        case _:
            print("Invalid Option")

elif(ser == 2):
    print("\nSelect Service\n\n1) Rent A Car\n2) Purchase Used Car")
    opt = int(input("\n(Type Option Number) : "))
    match(opt):
        case 1:
            print("\tCAR 1\n")
            p.Info()

            print("\n\tCAR 2\n")
            p1.Info()

            print("\n\tCAR 3\n")
            p2.Info()

            print("\n\tCAR 4\n")
            p3.Info()

            ans = int(input("\nSelect Car from following list (Type Number) : "))
            match(ans):
                case 1:
                    p = Rent("TOYOTA", "GRANDE", "2020",
                             "MANUAL", "Power", 15000)
                    p.Customer()
                    p.Charges()
                    print("\n\n~~~~~Reciept~~~~~\n\n")
                    p.Info()
                case 2:
                    p1 = Rent("NISSAN", "DAYZ", "2018",
```

```
                        "Auto", "Power", 110000)
                p1.Customer()
                p1.Charges()
                print("\n\n~~~~~Reciept~~~~~\n\n")
                p1.Info()
            case 3:
                p2 = Rent("SUZUKI", "WAGON-R", "2015",
                        "Auto", "Power", 100000)
                p2.Customer()
                p2.Charges()
                print("\n\n~~~~~Reciept~~~~~\n\n")
                p2.Info()
            case 4:
                p3 = Rent("HONDA", "CIVIC", "2022", "Auto", "Power", 20000)
                p3.Customer()
                p3.Charges()
                print("\n\n~~~~~Reciept~~~~~\n\n")
                p3.Info()

            case _:
                print("Invalid Option")

    case 2:
        print("\tWe have Following Cars\n\nCAR 1\n")
        s.Info()

        print("\n\tCAR 2\n")
        s1.Info()

        print("\n\tCAR 3\n")
        s2.Info()

        print("\n\tCAR 4\n")
        s3.Info()

        ans = int(input("\nSelect Car from above list (Type Number) : "))
        match(ans):
            case 1:

                s.Customer()
                print("\n\n~~~~~Reciept~~~~~\n\n")
                s.Reciept()

            case 2:

                s1.Customer()
                print("\n\n~~~~~Reciept~~~~~\n\n")
                s1.Reciept()
            case 3:

                s2.Customer()
                print("\n\n~~~~~Reciept~~~~~\n\n")
                s2.Reciept()

            case 4:
```

```
                    s3.Customer()
                    print("\n\n~~~~~Reciept~~~~~\n\n")
                    s3.Reciept()

                case _:
                    print("Invalid Option")
        case _:
            print("Invalid Option")
else:
    print("Invalid Option")
```