|  |  |  |
| --- | --- | --- |
|  | **THADOMAL SHAHANI ENGINEERING COLLEGE** |  |
| **DEPARTMENT OF INFORMATION TECHNOLOGY** |

**Roll no: I-62**

**6.OOPs: LO4**

**Aim:**

Design a system using classes for vehicles, rental agencies, and rental transactions. Implement methods to handle vehicle availability, rental periods, pricing, and customer

**Theory:**

* Python's classes and objects facilitate modular system design.
* Inheritance and encapsulation enable efficient management of vehicle availability and customer data.
* Methods are utilized to handle rental periods, pricing, and customer details.
* Polymorphism provides flexibility in managing various vehicle types.
* Exception handling addresses errors that may arise during rental transactions.

**Program:**

class Vehicle:

def \_\_init\_\_(self, vt, models, agency):

self.vt = vt

self.models = models

self.agency = agency

def display\_models(self):

print(f"\nAvailable {self.vt} from {self.agency} agency:")

for index, (model, details) in enumerate(self.models.items(), start=1):

print(f"{index}. {model}: Rent = Rs {details['rent']} per day, Number of Vehicles = {details['num\_vehicles']}")

def update\_model\_inventory(self, model\_name, rented\_quantity):

if model\_name in self.models:

self.models[model\_name]["num\_vehicles"] -= rented\_quantity

class RentalTransaction(Vehicle):

def \_\_init\_\_(self, vt, models, agency):

super().\_\_init\_\_(vt, models, agency)

def rental\_period(self):

while True:

try:

days = int(input("Enter the rental period (in days): "))

if days > 0:

return days

else:

print("The rental period must be a positive number. Try again.")

except ValueError:

print("Invalid input! Please enter a valid number.")

def process\_booking(self, model\_name, rental\_days, quantity):

if model\_name in self.models:

num\_vehicles\_available = self.models[model\_name]["num\_vehicles"]

rent\_price = self.models[model\_name]["rent"]

if quantity <= num\_vehicles\_available:

self.update\_model\_inventory(model\_name, quantity)

total\_amount = rent\_price \* quantity \* rental\_days

print(f"\n--- Booking Confirmation ---")

print(f"Your order for {quantity} {model\_name}(s) of {self.agency} agency is confirmed.")

print(f"Total rent for {rental\_days} days: Rs {total\_amount}")

print(f"Remaining Number of Vehicles for {model\_name}: {self.models[model\_name]['num\_vehicles']}")

else:

print(f"Sorry, only {num\_vehicles\_available} {model\_name}(s) are available right now.")

else:

print(f"{model\_name} is not a valid model.")

def main():

print(" Welcome to the Enhanced CAR and BUS Rental System! ")

print("-------------------------------------------------------------------------------")

car\_models = {

"Toyota Corolla": {"rent": 30, "num\_vehicles": 10},

"Honda Civic": {"rent": 40, "num\_vehicles": 8},

"Suzuki Swift": {"rent": 25, "num\_vehicles": 12},

"Toyota Fortuner": {"rent": 80, "num\_vehicles": 5},

"Ford Endeavour": {"rent": 90, "num\_vehicles": 4},

"Tesla Model X": {"rent": 150, "num\_vehicles": 3},

}

bus\_models = {

"Volvo Bus": {"rent": 100, "num\_vehicles": 6},

"Mini Bus": {"rent": 60, "num\_vehicles": 10},

}

car\_rental = RentalTransaction("Cars", car\_models, "Star Rentals")

bus\_rental = RentalTransaction("Buses", bus\_models, "Royal Bus Service")

while True:

print("\nMenu Options:")

print("1: View Available Cars")

print("2: View Available Buses")

print("3: Rent a Car")

print("4: Rent a Bus")

print("5: Exit")

choice = input("Enter your choice: ")

if choice == "1":

car\_rental.display\_models()

elif choice == "2":

bus\_rental.display\_models()

elif choice == "3":

car\_rental.display\_models()

try:

car\_choice = int(input("\nEnter the number of the car you want to rent: "))

if 1 <= car\_choice <= len(car\_rental.models):

model\_name = list(car\_rental.models.keys())[car\_choice - 1]

quantity = int(input(f"Enter the number of {model\_name}(s) to rent: "))

if quantity > 0:

rental\_days = car\_rental.rental\_period()

car\_rental.process\_booking(model\_name, rental\_days, quantity)

else:

print("Please enter a valid quantity.")

else:

print("Invalid choice! Please select a valid car number.")

except ValueError:

print("Invalid input! Please enter a valid number.")

elif choice == "4":

bus\_rental.display\_models()

try:

bus\_choice = int(input("\nEnter the number of the bus you want to rent: "))

if 1 <= bus\_choice <= len(bus\_rental.models):

model\_name = list(bus\_rental.models.keys())[bus\_choice - 1]

quantity = int(input(f"Enter the number of {model\_name}(s) to rent: "))

if quantity > 0:

rental\_days = bus\_rental.rental\_period()

bus\_rental.process\_booking(model\_name, rental\_days, quantity)

else:

print("Please enter a valid quantity.")

else:

print("Invalid choice! Please select a valid bus number.")

except ValueError:

print("Invalid input! Please enter a valid number.")

elif choice == "5":

print("Thank you for using our rental system. Goodbye!")

break

else:

print("Invalid choice! Please select a valid option.")

main()

**Output:**

Welcome to the Enhanced CAR and BUS Rental System!

-------------------------------------------------------------------------------

Menu Options:

1: View Available Cars

2: View Available Buses

3: Rent a Car

4: Rent a Bus

5: Exit

Enter your choice: 3

Available Cars from Star Rentals agency:

1. Toyota Corolla: Rent = Rs 30 per day, Number of Vehicles = 10

2. Honda Civic: Rent = Rs 40 per day, Number of Vehicles = 8

3. Suzuki Swift: Rent = Rs 25 per day, Number of Vehicles = 12

4. Toyota Fortuner: Rent = Rs 80 per day, Number of Vehicles = 5

5. Ford Endeavour: Rent = Rs 90 per day, Number of Vehicles = 4

6. Tesla Model X: Rent = Rs 150 per day, Number of Vehicles = 3

Enter the number of the car you want to rent: 2

Enter the number of Honda Civic(s) to rent: 7

Enter the rental period (in days): 8

--- Booking Confirmation ---

Your order for 7 Honda Civic(s) of Star Rentals agency is confirmed.

Total rent for 8 days: Rs 2240

Remaining Number of Vehicles for Honda Civic: 1

Menu Options:

1: View Available Cars

2: View Available Buses

3: Rent a Car

4: Rent a Bus

5: Exit

Enter your choice: 3

Available Cars from Star Rentals agency:

1. Toyota Corolla: Rent = Rs 30 per day, Number of Vehicles = 10

2. Honda Civic: Rent = Rs 40 per day, Number of Vehicles = 1

3. Suzuki Swift: Rent = Rs 25 per day, Number of Vehicles = 12

4. Toyota Fortuner: Rent = Rs 80 per day, Number of Vehicles = 5

5. Ford Endeavour: Rent = Rs 90 per day, Number of Vehicles = 4

6. Tesla Model X: Rent = Rs 150 per day, Number of Vehicles = 3

Enter the number of the car you want to rent: 1

Enter the number of Toyota Corolla(s) to rent: 11

Enter the rental period (in days): 9

Sorry, only 10 Toyota Corolla(s) are available right now.

Menu Options:

1: View Available Cars

2: View Available Buses

3: Rent a Car

4: Rent a Bus

5: Exit

Enter your choice: 4

Available Buses from Royal Bus Service agency:

1. Volvo Bus: Rent = Rs 100 per day, Number of Vehicles = 6

2. Mini Bus: Rent = Rs 60 per day, Number of Vehicles = 10

Enter the number of the bus you want to rent: 2

Enter the number of Mini Bus(s) to rent: 4

Enter the rental period (in days): 4

--- Booking Confirmation ---

Your order for 4 Mini Bus(s) of Royal Bus Service agency is confirmed.

Total rent for 4 days: Rs 960

Remaining Number of Vehicles for Mini Bus: 6

Menu Options:

1: View Available Cars

2: View Available Buses

3: Rent a Car

4: Rent a Bus

5: Exit

Enter your choice: 5

Thank you for using our rental system. Goodbye!

**Conclusion:**

OOPs (Object-Oriented Programming) in Python helps break down big systems, like vehicle rentals, into smaller, manageable parts. It makes the design easy to expand, fix, and reuse for future changes.