**Assignment 8**

**# Program 1**

class Car:

def \_\_init\_\_(self, registration\_number, max\_speed):

self.registration\_number = registration\_number

self.max\_speed = max\_speed

self.current\_speed = 0

self.travelled\_distance = 0

def main():

car = Car("ABC-123", 142)

print(f"Registration number: {car.registration\_number}")

print(f"Maximum speed: {car.max\_speed} km/h")

print(f"Current speed: {car.current\_speed} km/h")

print(f"Travelled distance: {car.travelled\_distance} km")

main()

**# Program 2**

class Car:

def \_\_init\_\_(self, registration\_number, max\_speed):

self.registration\_number = registration\_number

self.max\_speed = max\_speed

self.current\_speed = 0

self.travelled\_distance = 0

def accelerate(self, speed\_change):

self.current\_speed += speed\_change

if self.current\_speed > self.max\_speed:

self.current\_speed = self.max\_speed

elif self.current\_speed < 0:

self.current\_speed = 0

def main():

car = Car("ABC-123", 142)

car.accelerate(30) # Speed up by 30 km/h

car.accelerate(70) # Speed up by 70 km/h

car.accelerate(50) # Speed up by 50 km/h

print(f"Current speed: {car.current\_speed} km/h")

car.accelerate(-200) # Emergency brake

print(f"Final speed after emergency brake: {car.current\_speed} km/h")

main()

**# Program 3**

class Car:

def \_\_init\_\_(self, registration\_number, max\_speed):

self.registration\_number = registration\_number

self.max\_speed = max\_speed

self.current\_speed = 0

self.travelled\_distance = 0

def accelerate(self, speed\_change):

self.current\_speed += speed\_change

if self.current\_speed > self.max\_speed:

self.current\_speed = self.max\_speed

elif self.current\_speed < 0:

self.current\_speed = 0

def drive(self, hours):

self.travelled\_distance += self.current\_speed \* hours

def main():

car = Car("ABC-123", 142)

car.accelerate(60)

car.drive(1.5) # Drive for 1.5 hours at current speed

print(f"Travelled distance: {car.travelled\_distance} km")

main()

**# Program 4**

import random

class Car:

def \_\_init\_\_(self, registration\_number, max\_speed):

self.registration\_number = registration\_number

self.max\_speed = max\_speed

self.current\_speed = 0

self.travelled\_distance = 0

def accelerate(self, speed\_change):

self.current\_speed += speed\_change

if self.current\_speed > self.max\_speed:

self.current\_speed = self.max\_speed

elif self.current\_speed < 0:

self.current\_speed = 0

def drive(self, hours):

self.travelled\_distance += self.current\_speed \* hours

def main():

cars = []

for i in range(1, 11):

max\_speed = random.randint(100, 200)

car = Car(f"ABC-{i}", max\_speed)

cars.append(car)

race\_ongoing = True

while race\_ongoing:

for car in cars:

speed\_change = random.randint(-10, 15)

car.accelerate(speed\_change)

car.drive(1)

if car.travelled\_distance >= 10000:

race\_ongoing = False

break

print(f"{'Registration':<15}{'Max Speed':<15}{'Current Speed':<15}{'Travelled Distance':<20}")

print("="\*65)

for car in cars:

print(f"{car.registration\_number:<15}{car.max\_speed:<15}{car.current\_speed:<15}{car.travelled\_distance:<20.2f}")

main()

**# Program 5**

def main():

try:

balance = float(input("Enter account balance: "))

withdrawal = float(input("Enter withdrawal amount: "))

if withdrawal < 0:

raise ValueError("Withdrawal amount cannot be negative.")

if withdrawal > balance:

raise ValueError("Withdrawal amount exceeds balance.")

balance -= withdrawal

print(f"Withdrawal successful! New balance: {balance}")

except ValueError as e:

print(f"Error: {e}")

except Exception:

print("Invalid input. Please enter numeric values.")

main()

**# Program 6**

def write\_notes(filename):

note = input("Write a new note: ")

with open(filename, 'w') as file:

file.write(note + '\n')

print("Note written successfully.")

def append\_notes(filename):

note = input("Append a note: ")

with open(filename, 'a') as file:

file.write(note + '\n')

print("Note appended successfully.")

def read\_notes(filename):

try:

with open(filename, 'r') as file:

notes = file.readlines()

if notes:

print("Existing notes:")

for note in notes:

print(note.strip())

else:

print("No notes found.")

except FileNotFoundError:

print("No notes file found.")

def main():

filename = "notes.txt"

while True:

print("\n1. Write new notes")

print("2. Read existing notes")

print("3. Append notes")

print("4. Quit")

choice = input("Choose an option: ")

if choice == "1":

write\_notes(filename)

elif choice == "2":

read\_notes(filename)

elif choice == "3":

append\_notes(filename)

elif choice == "4":

print("Exiting the program.")

break

else:

print("Invalid option. Please try again.")

main()