Task 01

#include <iostream>

using namespace std;

class Car {

public:

string brand;

string model;

float rentalPrice;

bool availabilityStatus;

Car() {

brand = "Unknown";

model = "Generic";

rentalPrice = 0.0;

availabilityStatus = true;

}

void rentCar() {

if (availabilityStatus) {

cout << "Car rented successfully!" << endl;

availabilityStatus = false;

} else {

cout << "Sorry, the car already rented." << endl;

}

}

void displayCarInfo() {

cout << "Brand: " << brand << endl;

cout << "Model: " << model << endl;

cout << "Rental Price: $" << rentalPrice << endl;

cout << "Available: " << (availabilityStatus ? "Yes" : "No") << endl;

}

};

int main() {

Car myCar;

myCar.displayCarInfo();

myCar.rentCar();

myCar.displayCarInfo();

myCar.rentCar();

return 0;

}

TASK 02

#include <iostream>

using namespace std;

class Car {

public:

string brand;

string model;

float rentalPrice;

bool availabilityStatus;

Car(string b, string m, float price, bool status) {

brand = b;

model = m;

rentalPrice = price;

availabilityStatus = status;

}

void rentCar(int days) {

if (availabilityStatus) {

float totalPrice = rentalPrice \* days;

if (days > 10) {

totalPrice \*= 0.9;

} else if (days > 5) {

totalPrice \*= 0.95;

}

cout << "Car rented for " << days << " days. Total Price: $" << totalPrice << endl;

availabilityStatus = false;

} else {

cout << "Sorry, the car is already rented." << endl;

}

}

void displayCarInfo() {

cout << "Brand: " << brand << endl;

cout << "Model: " << model << endl;

cout << "Rental Price: $" << rentalPrice << endl;

cout << "Available: " << (availabilityStatus ? "Yes" : "No") << endl;

}

};

int main() {

Car car1("Toyota", "Corolla", 50.0, true);

car1.displayCarInfo();

car1.rentCar(7);

car1.displayCarInfo();

return 0;

}

TASK 03

#include <iostream>

using namespace std;

class Car {

public:

string brand;

string model;

float rentalPrice;

bool availabilityStatus;

Car(string b, string m, float price, bool status) {

brand = b;

model = m;

rentalPrice = price;

availabilityStatus = status;

}

Car(const Car &c) {

brand = c.brand;

model = c.model;

rentalPrice = c.rentalPrice;

availabilityStatus = c.availabilityStatus;

}

~Car() {

cout << brand << " " << model << " is removed from the system." << endl;

}

void displayCarInfo() {

cout << "Brand: " << brand << endl;

cout << "Model: " << model << endl;

cout << "Rental Price: $" << rentalPrice << endl;

cout << "Available: " << (availabilityStatus ? "Yes" : "No") << endl;

}

};

int main() {

Car car1("Honda", "Accord", 60.0, true);

car1.displayCarInfo();

Car car2 = car1;

car2.displayCarInfo();

return 0;

}

TASK 04

#include <iostream>

using namespace std;

class Car {

public:

string brand;

string model;

float rentalPrice;

bool availabilityStatus;

int rentalDays;

Car(string b, string m, float price, bool status) {

brand = b;

model = m;

rentalPrice = price;

availabilityStatus = status;

rentalDays = 0;

}

void rentCar(int days) {

if (availabilityStatus) {

rentalDays = days;

float totalPrice = rentalPrice \* days;

if (days > 10) {

totalPrice \*= 0.9;

} else if (days > 5) {

totalPrice \*= 0.95;

}

cout << "Car rented for " << days << " days. Total Price: $" << totalPrice << endl;

availabilityStatus = false;

} else {

cout << "Sorry, the car is already rented." << endl;

}

}

float calculateRevenue() {

return rentalPrice \* rentalDays;

}

void displayCarInfo() {

cout << "Brand: " << brand << endl;

cout << "Model: " << model << endl;

cout << "Rental Price: $" << rentalPrice << endl;

cout << "Available: " << (availabilityStatus ? "Yes" : "No") << endl;

}

};

int main() {

Car car1("Toyota", "revo", 55.0, true);

car1.displayCarInfo();

car1.rentCar(4);

car1.displayCarInfo();

cout << "Revenue Generated: $" << car1.calculateRevenue() << endl;

return 0;

}

TASK 05

#include <iostream>

#include <string>

using namespace std;

class Car {

string brand;

string model;

double rentalPrice;

bool available;

int regNumber;

double totalRevenue;

public:

Car(string b, string m, double price, int reg)

: brand(b), model(m), rentalPrice(price), regNumber(reg), available(true), totalRevenue(0.0) {}

void rentCar(int days) {

if (available) {

double totalPrice = rentalPrice \* days;

if (days > 10) {

totalPrice \*= 0.9;

} else if (days > 5) {

totalPrice \*= 0.95;

}

totalRevenue += totalPrice;

available = false;

cout << "Car rented for " << days << " days. Total price: $" << totalPrice << endl;

} else {

cout << "Car is not available for rent." << endl;

}

}

void returnCar() {

available = true;

cout << "Car returned successfully." << endl;

}

void displayDetails() {

cout << "Car Registration Number: " << regNumber << endl;

cout << "Brand: " << brand << endl;

cout << "Model: " << model << endl;

cout << "Rental Price: $" << rentalPrice << " per day" << endl;

cout << "Availability: " << (available ? "Available" : "Not Available") << endl;

cout << "Total Revenue Generated: $" << totalRevenue << endl;

}

};

int main() {

Car car1("Toyota", "Corolla", 50.0, 101);

Car car2("Honda", "Civic", 60.0, 102);

car1.displayDetails();

car1.rentCar(7);

car1.displayDetails();

car1.returnCar();

car1.displayDetails();

car2.displayDetails();

car2.rentCar(12);

car2.displayDetails();

return 0;

}