#include <iostream>

#include <vector>

#include <string>

using namespace std;

class Car {

private:

string make;

string model;

int year;

bool rented;

public:

Car(string make, string model, int year) {

this->make = make;

this->model = model;

this->year = year;

this->rented = false;

}

string getMake() { return make; }

string getModel() { return model; }

int getYear() { return year; }

bool isRented() { return rented; }

void setRented(bool rented) { this->rented = rented; }

};

class RentalSystem {

private:

vector<Car> cars;

public:

void addCar(Car car) {

cars.push\_back(car);

}

void displayAvailableCars() {

cout << "Available cars:\n";

for (Car car : cars) {

if (!car.isRented()) {

cout << car.getMake() << " " << car.getModel() << " " << car.getYear() << endl;

}

}

}

void rentCar(string make, string model, int year) {

for (Car& car : cars) {

if (car.getMake() == make && car.getModel() == model && car.getYear() == year && !car.isRented()) {

car.setRented(true);

cout << "Car rented successfully!\n";

return;

}

}

cout << "Sorry, the requested car is not available for rent.\n";

}

void returnCar(string make, string model, int year) {

for (Car& car : cars) {

if (car.getMake() == make && car.getModel() == model && car.getYear() == year && car.isRented()) {

car.setRented(false);

cout << "Car returned successfully!\n";

return;

}

}

cout << "Sorry, the requested car cannot be returned as it is not rented out.\n";

}

};

int main() {

RentalSystem rentalSystem;

Car car1("Toyota", "Corolla", 2020);

Car car2("Honda", "Civic", 2021);

Car car3("Ford", "Mustang", 2019);

rentalSystem.addCar(car1);

rentalSystem.addCar(car2);

rentalSystem.addCar(car3);

rentalSystem.displayAvailableCars();

rentalSystem.rentCar("Toyota", "Corolla", 2020);

rentalSystem.rentCar("Honda", "Civic", 2021);

rentalSystem.displayAvailableCars();

rentalSystem.returnCar("Toyota", "Corolla", 2020);

rentalSystem.displayAvailableCars();

return 0;

}

//End of code