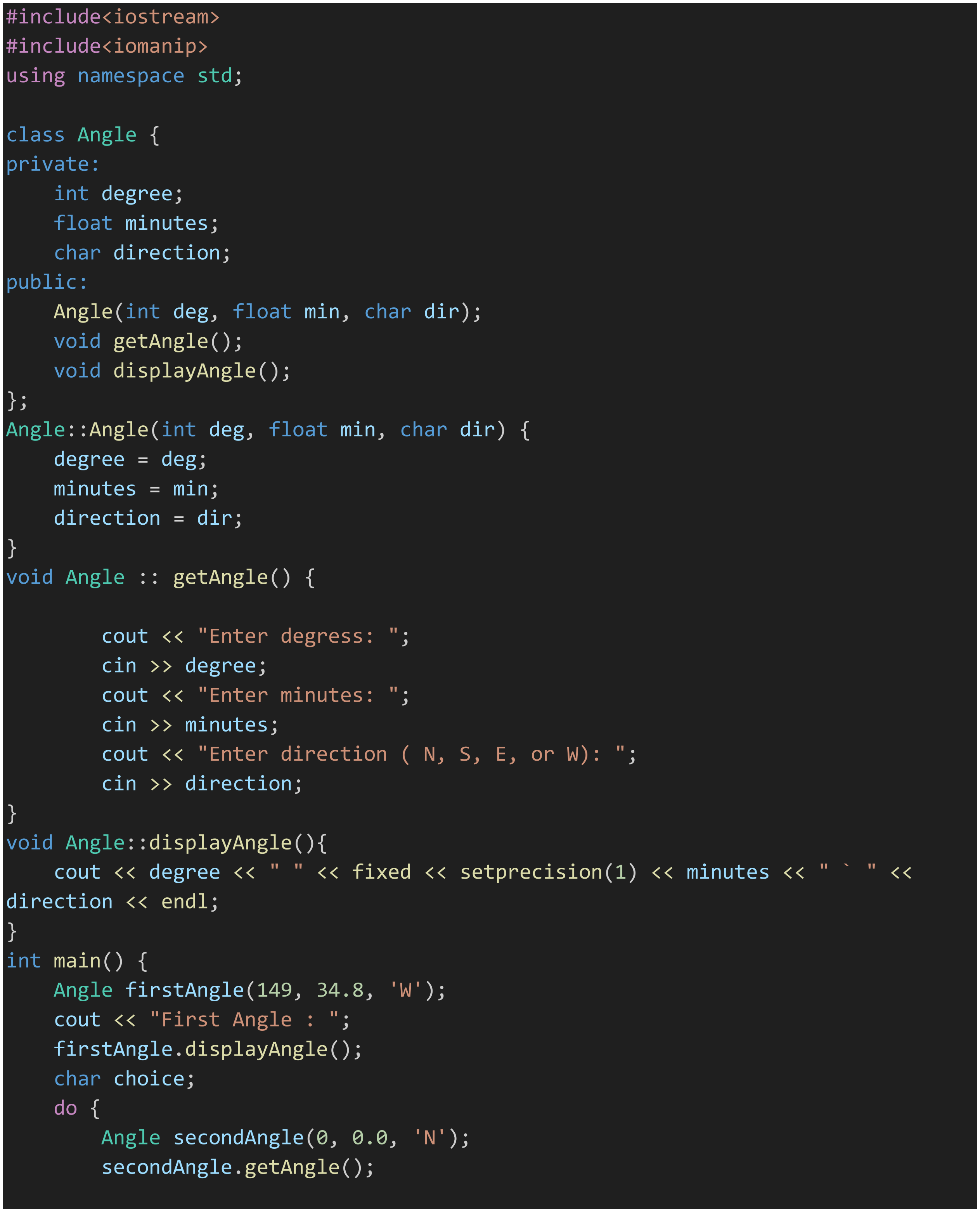
ASSIGNMENT 2

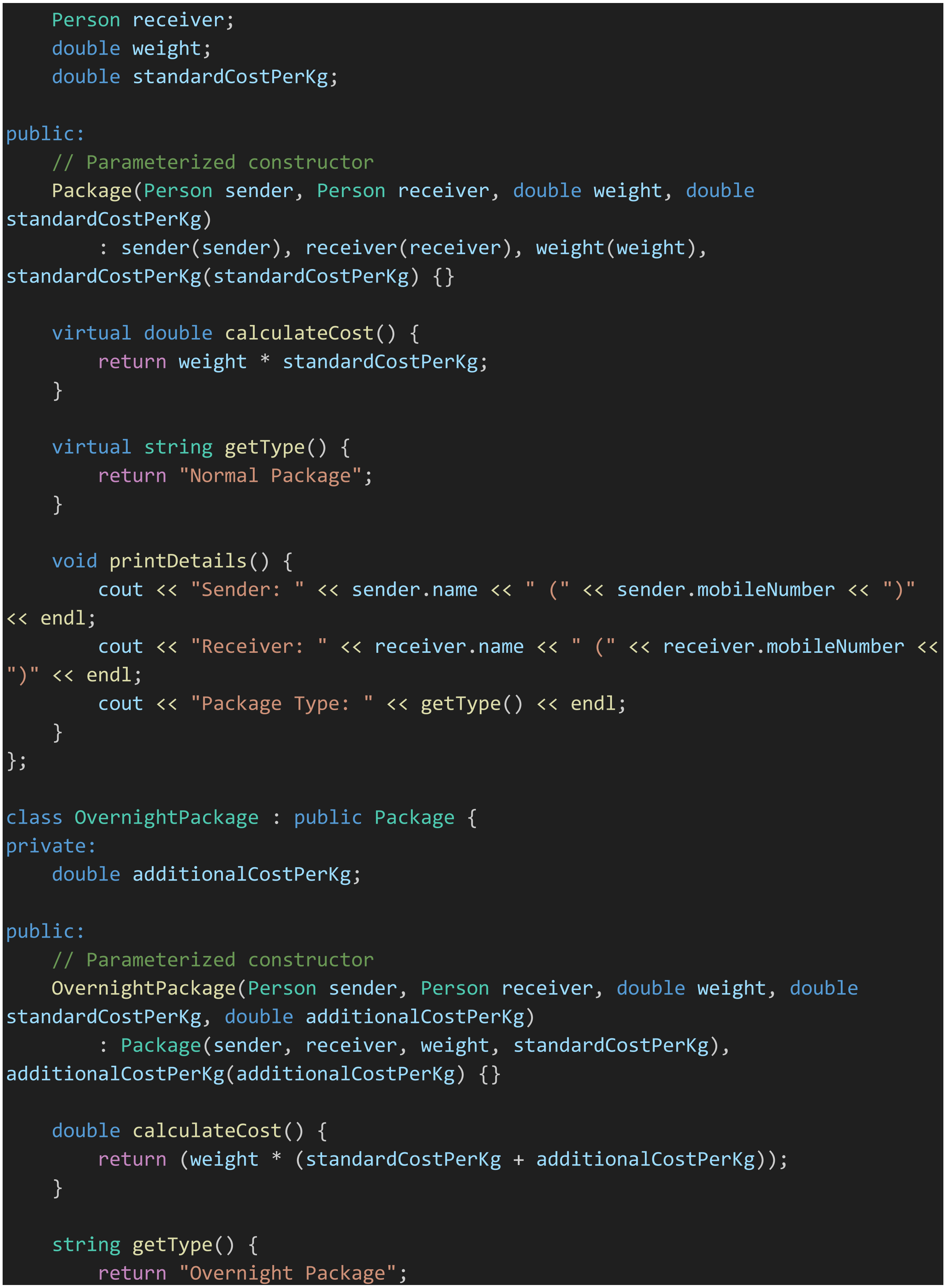
- **EX 1:**



```
#include<iostream>
#include<iomanip>
using namespace std;

class Angle {
private:
    int degree;
    float minutes;
    char direction;
public:
    Angle(int deg, float min, char dir);
    void getAngle();
    void displayAngle();
};
Angle::Angle(int deg, float min, char dir) {
    degree = deg;
    minutes = min;
    direction = dir;
}
void Angle :: getAngle() {

        cout << "Enter degress: ";
        cin >> degree;
        cout << "Enter minutes: ";
        cin >> minutes;
        cout << "Enter direction ( N, S, E, or W): ";
        cin >> direction;
}
void Angle::displayAngle(){
    cout << degree << " " << fixed << setprecision(1) << minutes << " ` " <<
direction << endl;
}
int main() {
    Angle firstAngle(149, 34.8, 'W');
    cout << "First Angle : ";
    firstAngle.displayAngle();
    char choice;
    do {
        Angle secondAngle(0, 0.0, 'N');
        secondAngle.getAngle();
        cout << "Entered Angle: ";
        secondAngle.displayAngle();

        cout << "You want to enter another angle (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');
}
```

- **Ex 2:**

```
#include <iostream>
#include <string>
using namespace std;

class Address {
public:
    string streetNumber;
    string suburb;
    string city;
    string province;

    Address(string streetNumber, string suburb, string city, string province)
        : streetNumber(streetNumber), suburb(suburb), city(city),
province(province) {}
};

class Person {
public:
    string name;
    string mobileNumber;
    Address address;
    //parametrized constructor
    Person(string name, string mobileNumber, Address address)
        : name(name), mobileNumber(mobileNumber), address(address) {}
};


class Package {
protected:
    Person sender;
```

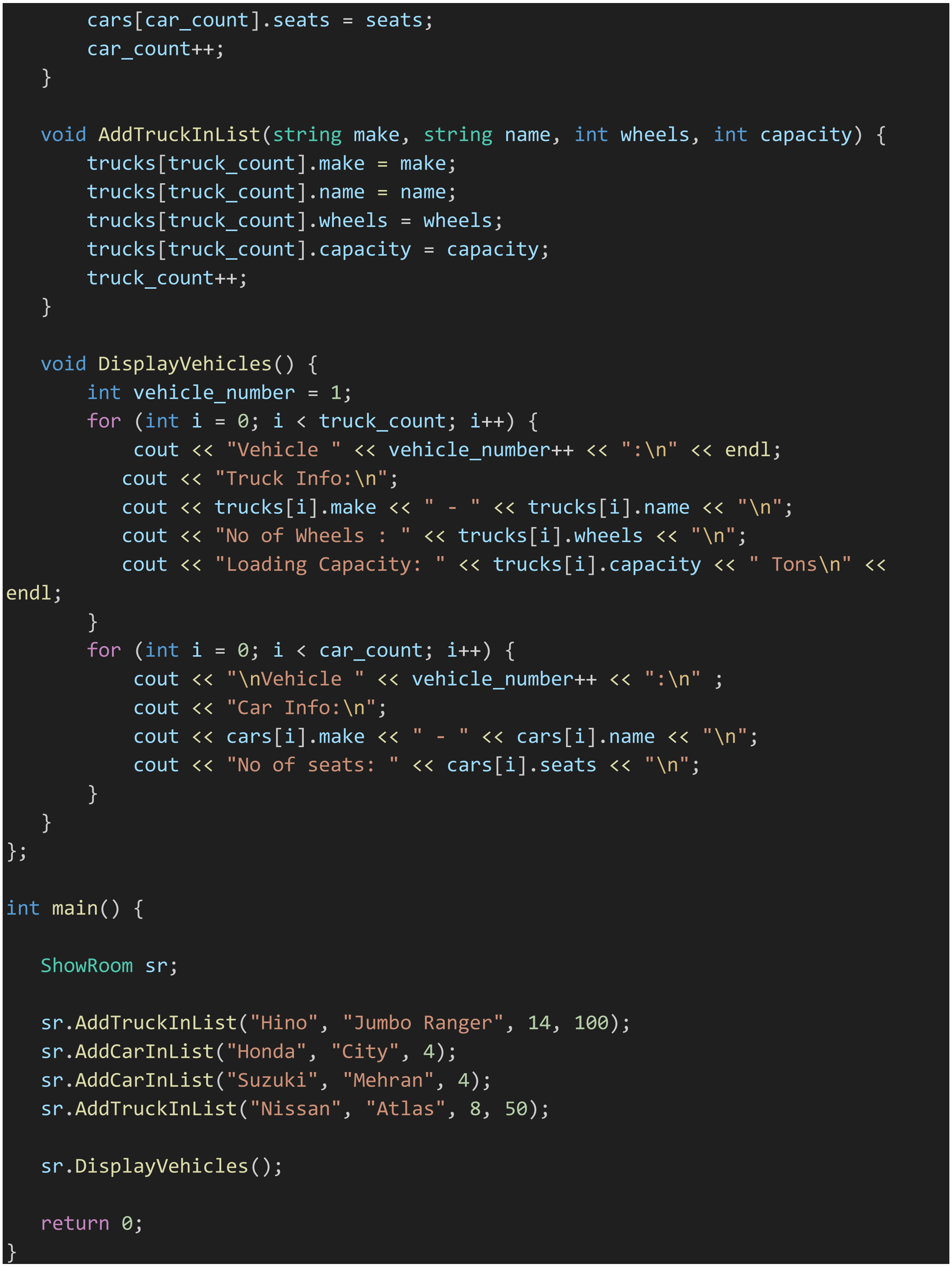


```
    Person receiver;
    double weight;
    double standardCostPerKg;

public:
    // Parameterized constructor
    Package(Person sender, Person receiver, double weight, double
standardCostPerKg)
        : sender(sender), receiver(receiver), weight(weight),
standardCostPerKg(standardCostPerKg) {}

    virtual double calculateCost() {
        return weight * standardCostPerKg;
    }

    virtual string getType() {
        return "Normal Package";
    }

    void printDetails() {
        cout << "Sender: " << sender.name << " (" << sender.mobileNumber << ")"
<< endl;
        cout << "Receiver: " << receiver.name << " (" << receiver.mobileNumber <<
")" << endl;
        cout << "Package Type: " << getType() << endl;
    }
};

class OvernightPackage : public Package {
private:
    double additionalCostPerKg;

public:
    // Parameterized constructor
    OvernightPackage(Person sender, Person receiver, double weight, double
standardCostPerKg, double additionalCostPerKg)
        : Package(sender, receiver, weight, standardCostPerKg),
additionalCostPerKg(additionalCostPerKg) {}

    double calculateCost() {
        return (weight * (standardCostPerKg + additionalCostPerKg));
    }

    string getType() {
        return "Overnight Package";
    }
};

class TwoDayPackage : public Package {
private:
    double flatFee;

public:
    // Parameterized constructor
    TwoDayPackage(Person sender, Person receiver, double weight, double
standardCostPerKg, double flatFee)
        : Package(sender, receiver, weight, standardCostPerKg), flatFee(flatFee)
{}

    double calculateCost() {
        return (weight * standardCostPerKg) + flatFee;
    }

    string getType() {
        return "Two-Day Package";
    }
};

int main() {
    // Array of pointers to Package objects
    const int numPackages = 3;
    Package* packages[numPackages];

    // Create sender and receiver objects
    Person sender("Ali", "03211234567", Address("312", "Abubakar Block, Garden
Town", "Lahore", "Punjab"));
    Person receiver("Saheen", "03123456789", Address("123", "Gulshan Block ",
"Karachi", "Sindh"));

    // Instantiate different types of packages
    packages[0] = new Package(sender, receiver, 5.5, 10.0);
    packages[1] = new OvernightPackage(sender, receiver, 3.0, 12.0, 5.0);
    packages[2] = new TwoDayPackage(sender, receiver, 2.0, 8.0, 20.0);

    // Loop through the array and print details
    for (int i = 0; i < numPackages; i++) {
        cout << "Package " << i + 1 << " details:" << endl;
        packages[i]->printDetails();
        cout << "Total delivery cost: " << packages[i]->calculateCost() << endl
<< endl;
```

```
    }

    // Free memory allocated for the packages
    for (int i = 0; i < numPackages; i++) {
        delete packages[i];
    }

    return 0;
}
```

Ex 3:

```
#include<iostream>
#include<string>
using namespace std;

class Vehicle {
public:
  string make;
  string name;
};

class Car : public Vehicle {
public:
   int seats;
};

class Truck : public Vehicle {
public:
   int wheels;
   int capacity;
};

class ShowRoom {
public:
   Car cars[100];
   Truck trucks[100];
   int car_count = 0;
   int truck_count = 0;

   void AddCarInList(string make, string name, int seats) {
       cars[car_count].make = make;
       cars[car_count].name = name;
```



```
       cars[car_count].seats = seats;
       car_count++;
   }

   void AddTruckInList(string make, string name, int wheels, int capacity) {
       trucks[truck_count].make = make;
       trucks[truck_count].name = name;
       trucks[truck_count].wheels = wheels;
       trucks[truck_count].capacity = capacity;
       truck_count++;
   }

   void DisplayVehicles() {
       int vehicle_number = 1;
       for (int i = 0; i < truck_count; i++) {
           cout << "Vehicle " << vehicle_number++ << ":\n" << endl;
          cout << "Truck Info:\n";
          cout << trucks[i].make << " - " << trucks[i].name << "\n";
          cout << "No of Wheels : " << trucks[i].wheels << "\n";
          cout << "Loading Capacity: " << trucks[i].capacity << " Tons\n" <<
endl;
       }
       for (int i = 0; i < car_count; i++) {
           cout << "\nVehicle " << vehicle_number++ << ":\n" ;
           cout << "Car Info:\n";
           cout << cars[i].make << " - " << cars[i].name << "\n";
           cout << "No of seats: " << cars[i].seats << "\n";
       }
   }
};

int main() {

   ShowRoom sr;

   sr.AddTruckInList("Hino", "Jumbo Ranger", 14, 100);
   sr.AddCarInList("Honda", "City", 4);
   sr.AddCarInList("Suzuki", "Mehran", 4);
   sr.AddTruckInList("Nissan", "Atlas", 8, 50);

   sr.DisplayVehicles();

   return 0;
}
```