Car Rental System

Aya Atef Mahmoud

Section:5

ID:1158

Introduction:

Car rental system is a software that organizes all operations on car rental including managing reservation operations , payment either to pay money in full or in installments.

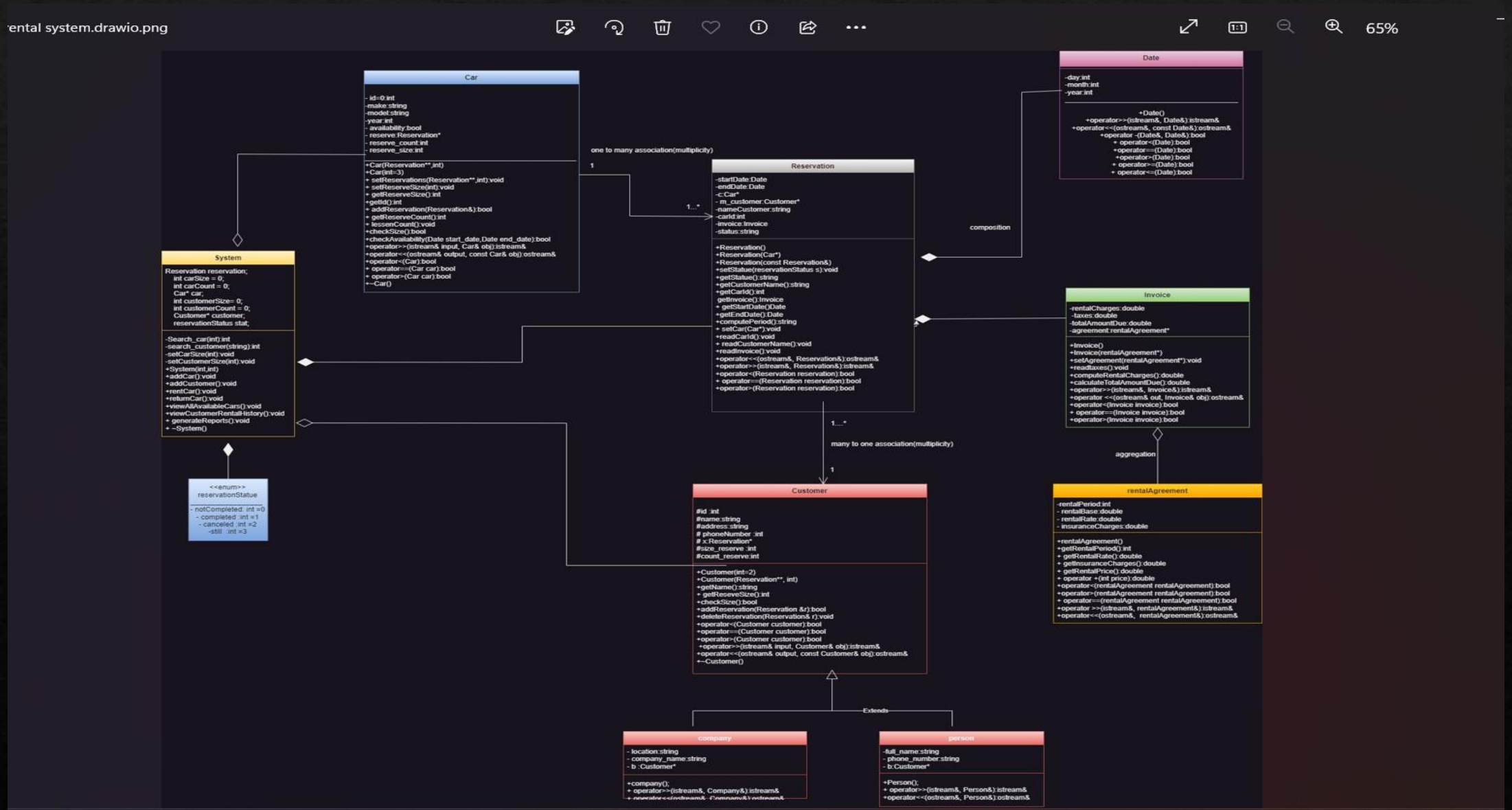
It provides the user with information of the car he is going to renter including the place where it made , the model of the car ,the year when it was made ,the id of the car and if it is available to renter at this time or not.

A car can be rented several times but there musn't be overlapping as a customer can't renter a car which is rented by another customer

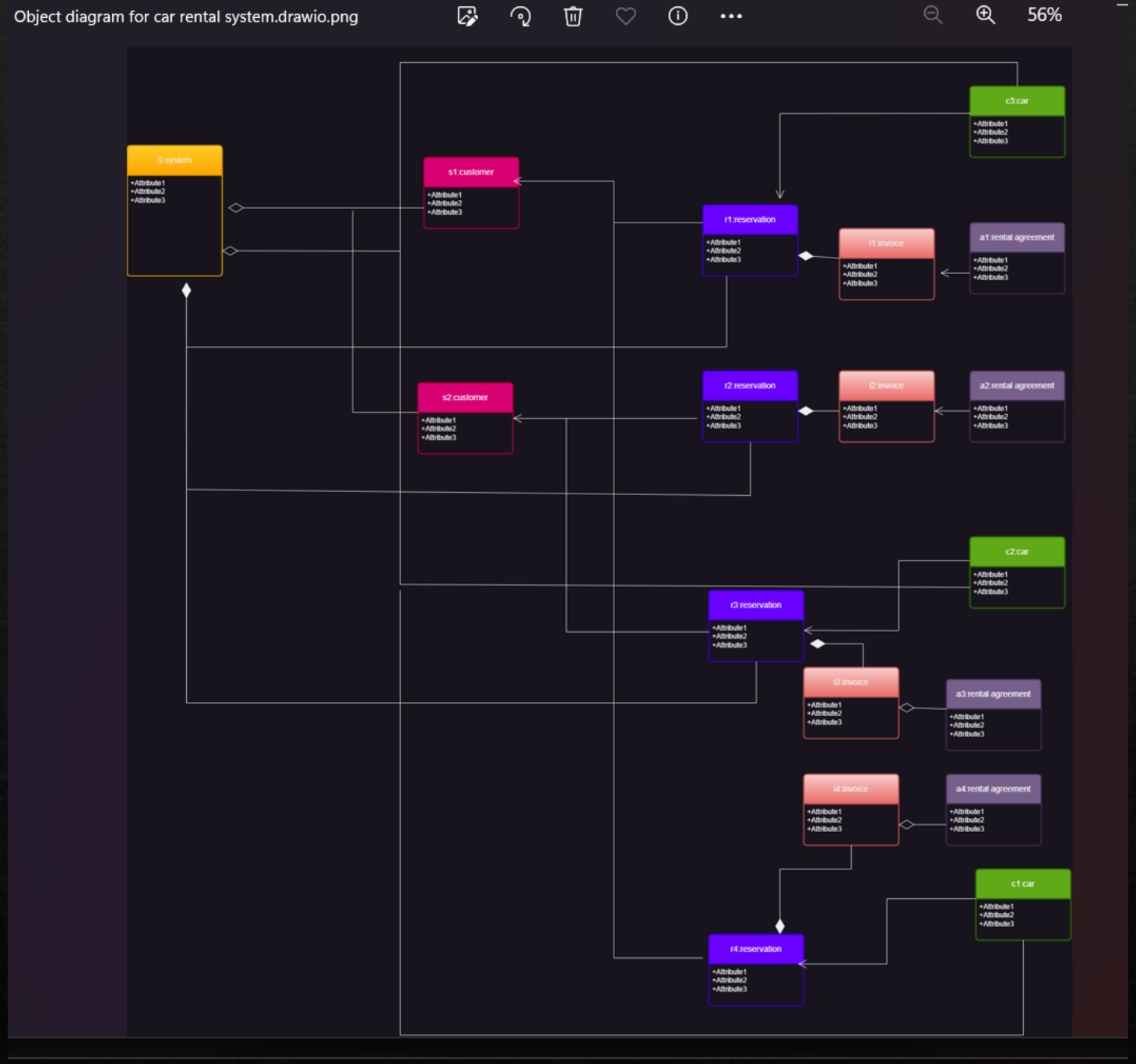
A customer also can rent several times ,he can either rent the same car or another one but in different times

The system then manages reservations of cars to employees

Classes Diagram:



Object Diagrams



A tour around functions

*We will start with class Date:

Attributes:

Day,month ,Year are the main components of any date

Funcfions:

I overloaded the operator insertion and extraction in order to read data with the keyword cin and display them with the key word cout

Also I overloaded the operator – in order to compute the period of renting the car

And the operators of comparison like >,<,== in order to compare between objects of the date and see if there is overlapping or not

The operator – works like the following:

First it suptracts the years together and it is normal that the second year is greater than the first one ,Next it subtracts months together and if the first one is less than the second one it converts one year into 12 months and then do the usual subtraction

Third it subtracts days together and if the result is negative we convert one month into 30 day and do the usual subtraction

For example:
If the first date is 12/3/2023
Second date is:13/5/2022
So how can we subtract them
We say 2023-2022=1 year
3-5 =-2 #isn't allowed
So we convert year to months and subtract
1*12+3-5=10 months
Then we subtract days
12-13=-1#isn't allowed
Then we convert one month to 30 day and subtract as normal
30*1+12-13=29 day
Then the result is 9 months and 29 day

The function overlap :
It isn't a member function but it is a friend function and I takes two parameters of type date .
I made it a friend one not part of the class as when I call it I needn't to call it by the object of its class then put this object also as a parameter to the function
Then what did I do in this function what is the implementation of it?

The implementation of the function is as follows:

overlapping means that two periods intersects with each other or one period consist of the other or both the periods are identical

then how can I implement this in coding

as we know that a car is rented then the when another user comes to renter it the start date of him is after than the start date of the first person rented it

so then intersection happens if the start date of this person is between the first and start date of the first person

that means that the start date of the second person Is greater than or equal to the start date of the first person and less than or equal to the end date of the first person ignoring the position of the end date of the second person as the first date is sufficient to insure overlapping.

Then the idea of the code that it checks if the start date of the first person is less than or equal to the end date of the first person

The rational operators compare dates together

It is all about class date…..

Class car:

it is a dependent instance in the system it is the main component of the system and it has the main attributes like the id the name of the place where it was made ,the model of the car and the availability of the car to be rented

it also has a pointer to the class reservation but why pointers ?

I didn't make it as an object as the life time of the reservation isn't dependent on the life time of the car also the car doesn't create the reservation it is the system which creates the reservation another thig is reservation isn't part of car only car uses this class o ensure that some one rented a car at the right time ,because of this reason I used the association relationship neither the composition nor the aggregation.

The pointer points to an array of reservations as a car can be rented several times but a reservation can't have more than a car as it is special for one person and one car

I also have the reserve_size to ensure the maximum size of reservations and the reserve_count to see how many reservations are there in my system

Functions:

I have made two parametrized constructors and the both are used in different cases

the first one receives an array of cars this array can be defined in the main menu and the size of the array

so what I did in this function

I assigned the reserve_size to the size which is given as a parameter

then I allocated memory for the array defined in the class in heap to ensure that they are allocated in heap and made a deep copying of the array given as a parameter to the one defined in my class

and I made deep copying to insure all the attributes and functionality f the array given as a parameter to the one defined in my class

the deep copying is done by the copy constructor

The second parametrized constructor is the one which takes the size as a parameter and allocate my array in the heap with this size
I made this function to allow the dealt between the system and the user as I made a function to add reservation that asks the user the to enter the details of the reservation then add this to the array with the increase in the count of reservations allocated in my array

---

The setReservations: does the same functionality as the parametrized constructor with the two parameters do

---

setresevesize: check if the size entered is valid or not by checking if it is positive or negative

---

The function getResereSize :returns the size of the array reservation

---

The function addReservation: first it checks if there is place to add more reservations if so it assigns the one given by reference to the one defined in the class so we can now store it in our array

---

lessenCount:lessen the cout of my array and I used it when I return a car I lessen then the number of reservations and I can't access the count outside the class directly as it is declared in the private section

---

boolcheckSize():This function is used to check if I didn't exceed the max size so it allows me outside the class to check if I can add more reservations or not

---

bool checkavailability(): this function checks if the car is available and it receives two parameters for the start and end date it first call the function overlap to check if there is overlapping then if the function return true this means there is overlapping and the car isn't available in this period so the availability is equal to zero else there isn't overlapping and the car is available for renting

---

Operator>>	I overrided the operator >> for insertion the return type is istream class and I returned it by reference ,the function is made a friend to the class and it allows the user to enter the details of the car like :id,model,make,the year when it was made
Operator<<	The return type is the reference of ostream class and it takes two parameters one is of ostream class passed also by reference and the reference of class Car which allows me to display the details of the car which was entered also to print the array of reservations if there is any and if not it won't print any thing
Operator>	To check if the size is greater than the size of actual reservations stored
Operator <	Checks if the size of the array is less than the actual stored
Operator==	Checks if they are equal or nt
~Car()	The destructor is used to delete what is dynamically allocated in my array
Reservation(Reservation&)	I defined The copy constructor of reservation here as I included the header file here but I can't include the header file of class car in class reservation as it will cause circular dependency and it will result in infinity loop and the program linker would give me unexpected errors and in the second view I want to access the attributes of my car to provide the equality of the pointer car in order to enhance circular deep copying and avoid unexpected behaviors of shallow copying like in the deletion it may give heap corruption as the two pointers refer to the same memory address but in deep copying each pointer has its

Class Customer:

it has the private attributes of the customer like name , address ,phone number and a pointer to array of reservations as a customer can reserve more than a car in different times
also the size_reserve to determine the utmost size of the array and the size_count that has the actual number of reservations to an employee
a customer can be a person or a company so here we use generalization to generalize their attributes in the base class(customer) but special attributes are specialized to the derived ones

Functions:

Customer(int)	The parametrized constructor initializes the value of the size
Customer(Reservation**,int)	Receives a pointer to an array if it is defined in the main function allocating memory for my array to avoid crashes or unexpected behaviors with size given as a parameter The loop iterates around the array to check if every element in my array is assigned to which is given as a parameter The initializing was by using the copy constructor to ensure every element is initialized correctly
String getName()	Returns the name of the customer and I defined it to use it when searching for the customer
Int getReserveSize()	Returns the size of reservations
Bool checkSize()	Check if the count exceeded the size or not
dReservation(&Reservation)	Checks if the array is full and if not it adds the reservation given by reference to it then return s true
Void teReservation(&Resevation)	It takes the reference of the reservation given as a parameter then loops over the array if found delete it from the array and lessen the count

Operator<	Checks if the reservation count of the object is less than the one given as parameters
Operator>	Checks if the reservation count of the object is greater than the one given as parameters
Operator==	Check if they are equal or not
istream& operator>>(istream& input, Customer& obj)	To allow entering the values of the customer like name , address , phone number
ostream& operator<<(ostream& output,const Customer& obj)	To allow for the output of the customer details also print array of reservations if found
~Customer()	Deletes the array of reservations

Class Rental agreement:

it is responsible for the agreement of payment including rental period , rental base , insurance charges

rentalAgreement()	Default constructor to initialize the starting vaiues of the attributes
Double getRentalPeriod	Return the rental period
Double getRentalRate()	Returns the rental rate
Double getInsuranceCharges()	Returns insurance charges
Double getRentalPrice()	Returns the rentalRate*RentalPeriod+insuranceCharges as the rental price of the car
The operator+	It adds more money .it pays the rest of money required for the car
Operator<	Checks if the rental price is less than the rental price of another object
Operator>	Checks if the rental price is greater than the rental price of another object

Operator ==	Checks if the rental price is equzl to the rental price of another object
Operator>>	To allow inserting data of the class rental agreement and this is a friend function the return type of is of type istream and is returned by reference and has two parameters one of object istream and the other of object rentalAgreement
Operator<<	Allow extracting the data of the class and it is a friend function to the class not a member function its return type is ostring& and takes two arguments one of type ostream& and the other of type rentalAgreement& and it is preferred to make it constant as we won't modify anything we only read data

Class invoice :

the attributes are :rental Charges , Taxes, rentalAmountDue
also a pointer to the rental agreement as it has details needed for the rental invoice and the relation is

Class functions

Invoice()	The default cinstructor is for initializing dtata members with zero also allocating memory for the place the pointer supposed to point to
Invoice(rentalAgreement*)	Setting the pointer to another pointer supposed to be defined in the main function
Void setAgreement()	Does the same functionality as the parametrized constructor
Void readTaxes()	Asks the user to enter the taxes and allow him to add them
Double computeRentalCharges()	This functions calculates all the charges would be paid for the car including taxes then it adds to the total rental price the value of taxes
doubleCalculateTotalAmountDue()	
Operator>>	A friend function to the class allows the user to enter the dtat required first it enters the data of class rentalAgreement to not repeat codes then allows insertion for the taxes
Operator<<	A friend function to the class allows the extraction of the data into the object out of the ostream class and takes another argument is of type Invoice .it is passed by reference but not const as we called the function <<of the retalAgreement object and it isn't const function so

Operator<	Checks if the totalAmountDue is less than the one in another object
Operator>	Checks if the totalAmountDue is greater than the one in another object
Operator==	Checks if the totalAmountDue is equal to the one in another object

Class reservation:

it is responsible for the customer to reserve and for a car to be reserved

The attributes:

start date , end date , id of a car and name of a customer

An object of invoice and pointer to car and employee as they are now instances of out system and their life time isn’t dependent .it has an object of type invoice here I used composition as there is no meaning for invoice without the class reservation so

The life time is dependent and one reservation has one invoice and invoice is owned by one so the relation is one to one composition

I also defined an enum called reservation Statue to se the status of the reservation

class functions:

Reservation()	The default constructor is used to initialize values
Reservation(car*p)	Parametrized constructor to make the car in my reservation points at the same place as the pointer given has a parameter
Void setStatus(string s)	Sets the status of my reservation to the one given as a parameter
string getStatus()	Returns the status of my reservation
String getCustomerName()	Returns the name of the customer as a string as I needed it in search for the customer in my system in order to check the reservation is valid or not
Int getId()	Returns the id of the car and I used it to search for the car in my system too validate the reservation
Invoice getInvoice()	Returns the invoice of the reservation
Void setCar(car*)	Do the sane functionality as the parametrized constructor using the pare assignment to compare the objects

Date getSartDate()	Returns the start date of reservation
Date getEndDate()	Returns the end date of the reservation and I used these two functions when comparing to the start date and end date of another object to check is there is overlapping or not
String computePeriod()	It subtracts the start date from the end date and the result is stored in a string in the form day/month/year
Void readCarId()	Allows the user to enter the id of the car he want to reserve
Void readCustomerName()	Allows the user to enter his name in order to insure the operation of reservation
Void readInvoiceDetails()	Allows entering the details of the invoice with the operator >> as I overloaded it in my function and I made this function and didn't allow entering its details with the details of the reservation as in the future when I wan to reserve a car first check if the car specified and the customer found or not then asks to enter the start date and end date of the reservation to check if there is overlapping or not then if every thing is ok I enter the money I would pay for the reservation
The operator>>	Allows to enter the start date and end data of the reservation

The operator<<	Allows to extract the startdate and the end date then reserve them in the out object in ot=rder if I want to print them in the secreen I use th object of ostream(cout)to print these details in my screen
Operator<	Checks if the total amount due is less than the other one
Operator>	Checks if the total amount due is greater than the other one
Operator==	Checks if the total amount due is equal to the other one

Class system:

this class id the one which will be the interface of my program .this class has the control of all what happens in my system and manages it .This class manages the operations of adding a car or adding a customer ,renting a car , returning a car ,view all available cars , view rental history and generating reports of cars and persons

its attributes are a pointer to an array of cars ,the utmost size of cars in the system ,the count of cars in the system ,a pointer to an array of customers , the utmost size of the array of customers and the count of them

The functions:

first I defined the functions search_car(),search_customer(),setCarsize(),setCuistomerSize() in the private section as it uses them as helper functions and they are called

System(int,int)	Takes two parameters the size of the array of the customers and the size of the array of cars,it checks if the size given is valid or not by setters then it allocates the two arrays in the memory with the ize given if valid
Void setCarSize(int)	Checks if the size of cars given as a parameter is valid or not by checking if it is positive or negative
Void setCustomerSize(int)	Checks if the size of customers is valid or not by checking if it is positive or negative
Int search_car(int)	It has the id given as a parameter , it loops around the array of cars to check if the id passed as a parameter equals the id of the car of some index if so it returns its index ,if not it checks the other index and if not found in the array it returns -1
Int search_customer(string)	It searches in the array for the customer by its name if found it returns his index if not go to the other index and check if we reached the end of the array and didn't find this customer it returns -1
Void addCar()	First it checks if there is place in the array to add cars so as not cause buffer overflow or out_of_bound error hen if I have space in my array to add more elements the functions allows the user to enter details of the car then make the array of the index count equal this car and then increases the size by one this is done inside a loop so as if I want to add more than a car I can do it by entering y in every time then enters n if there is enough
Void addCustomer()	Does the same functionalities as the function addCar

Void rentCar()	**It is responsible for renting a car this function first search for the car by id ,if not found it exists the function but if found it searchs for the customer by name ,if not found it exists the function but if found then asks the user to enter the details of the details of the reservation these details are the start date and the end date and loops over the reservations of the car to see if there is overlapping then finding overlapping makes the car not available at that time which causes exiting the function but if there isn't overlapping this causes the car be available in this period and then the reservation comes true and we add it for both the array of cars and the array of customers**
Void return a car ()	After the period of reservation finishes the customer needs to return the car so we delete the reservation of the car from the array and then the car is available for any other reservations
Void viewAllAvailableCars( )	You enter the start and the end date of the reservation and there is a loop that loops over the reservation of all cars to check if the car can be available in this period and what car can be available
Void generateReports()	It loops over the array of cars and print them also the reservations in them Also it loops over the customer array to print him also his array of reservations
Void view customer rentalHistory()	

~ScreenShots for the code~

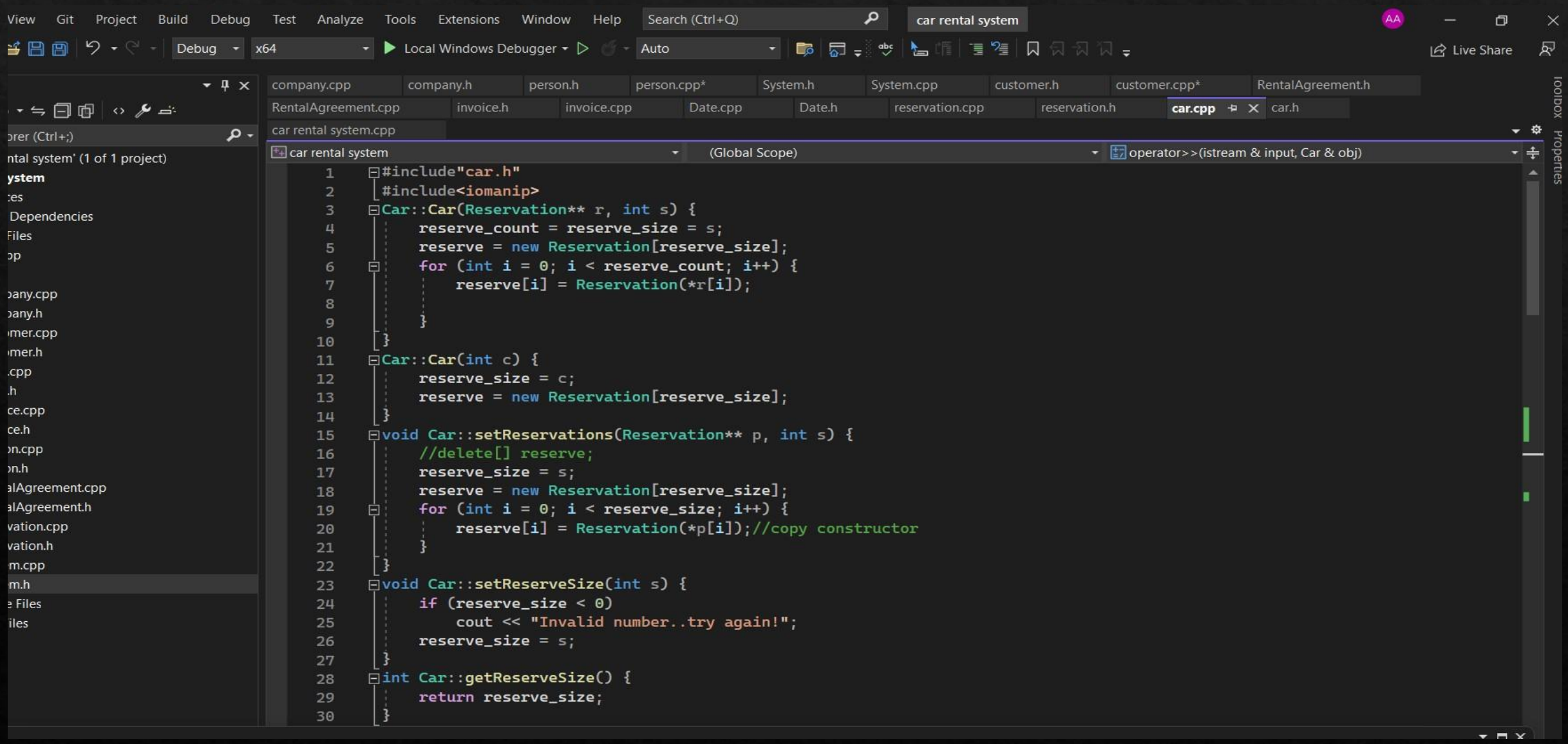
header file of car

oject)

car rental system | Car | getReserveCount()

```
1   #pragma once
2   #include<iostream>
3   #include"reservation.h"
4   using namespace std;
5   class Car {
6       int id=0;
7       string make;
8       string model;
9       int year=0;
10      bool availability=1;
11      Reservation* reserve=NULL;
12      int reserve_count=0;
13      int reserve_size=0;
14  public:
15      Car(Reservation**,int);
16      Car(int=3);
17      void setReservations(Reservation**,int);
18      void setReserveSize(int);
19      int getReserveSize();
20      int getId();
21      bool addReservation(Reservation&);
22      int getReserveCount();
23      void lessenCount();
24      bool checkSize();
25      bool checkAvailability(Date start_date,Date end_date);
26      friend istream& operator>>(istream& input, Car& obj);
27      friend ostream& operator<<(ostream& output, const Car& obj);
28      bool operator<(Car);
29      bool operator==(Car car);
30      bool operator>(Car car);
31      ~Car();
32  };
33  bool overlap(Date s1, Date d1, Date s2, Date d2);
34
```

Source file for class car



company.cpp | company.h | person.h | person.cpp* | System.h | System.cpp | customer.h | customer.cpp* | RentalAgreement
RentalAgreement.cpp | invoice.h | invoice.cpp | Date.cpp | Date.h | reservation.cpp | reservation.h | car.cpp | car.h
car rental system.cpp

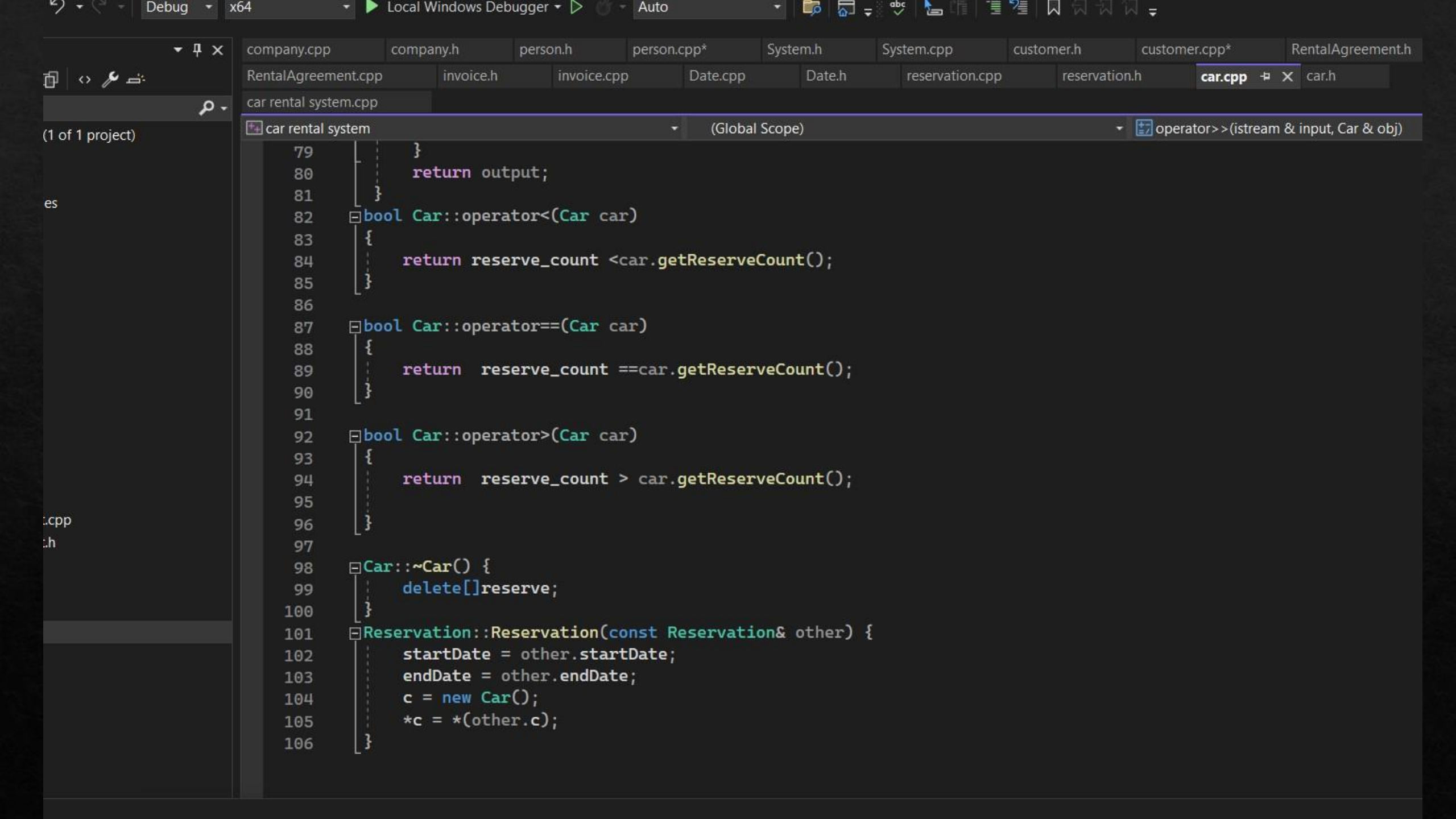
car rental system | (Global Scope) | operator>>(istream & input, Car & obj)

```
31  int  Car::getId(){
32      return id;
33  }
34  bool Car::addReservation(Reservation &r) {
35      if (reserve_count < reserve_size) {
36          reserve[reserve_count++] = r;
37          return true;
38      }
39      else
40          cout << "You exceeded the utomst number of cars ";
41      return false;
42  }
43  int Car::getReserveCount() {
44      return reserve_count;
45  }
46  void Car::lessenCount() {
47      reserve_count--;
48  }
49  bool Car::checkSize() {
50      return reserve_count <= reserve_size;
51  }
52  bool Car::checkAvailability(Date start_date, Date end_date)
53  {
54      for (int i = 0; i < reserve_count; i++)
55      {
56          if (overlap(reserve[i].getStartDate(), reserve[i].getEndDate(), start_date, end_date)) {
57              availability = 0;
58              return false;
59          }
```

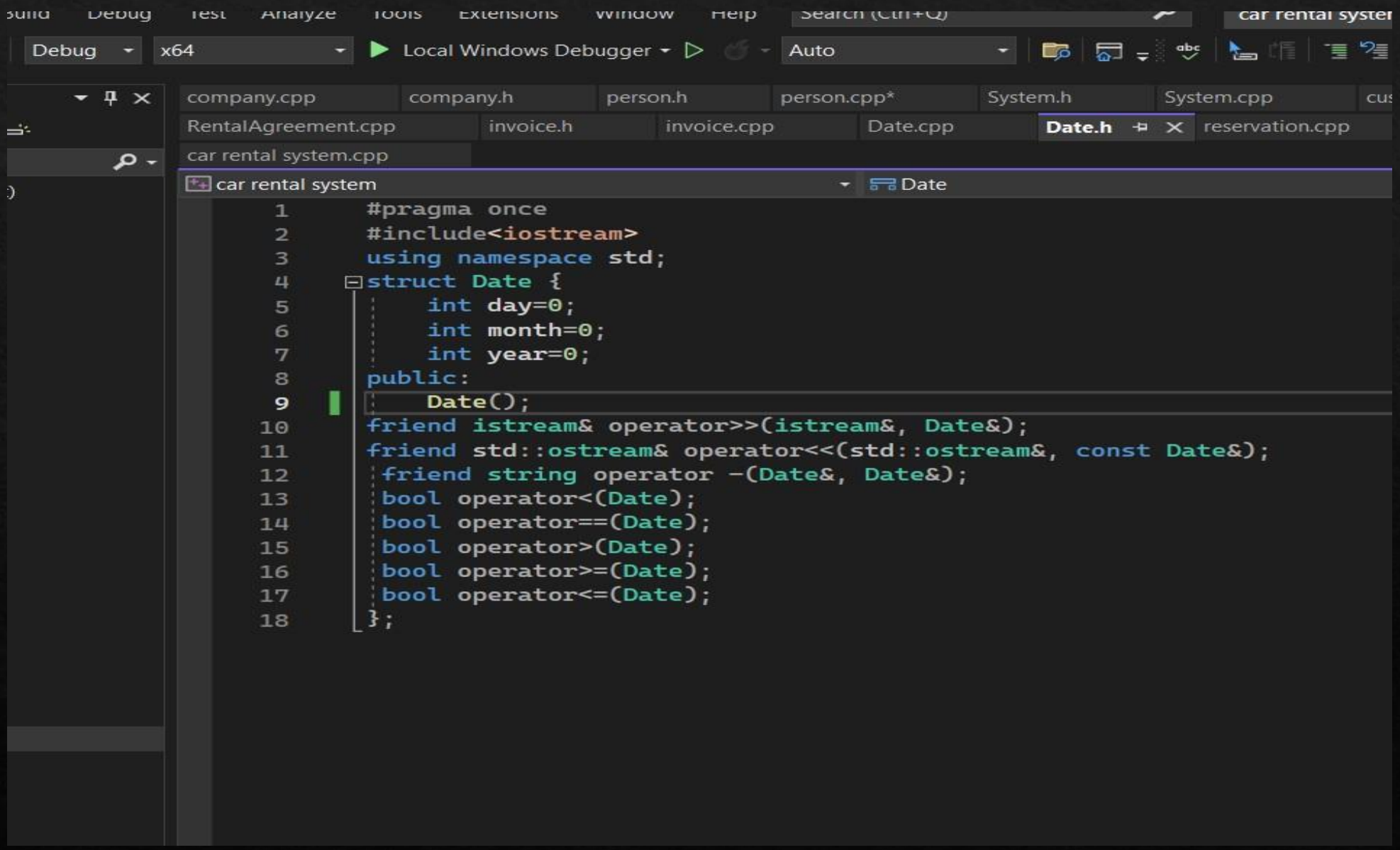
company.cpp | company.h | person.h | person.cpp* | System.h | System.cpp | customer.h | customer.cpp* | RentalAgreement.h
RentalAgreement.cpp | invoice.h | invoice.cpp | Date.cpp | Date.h | reservation.cpp | reservation.h | car.cpp | car.h
car rental system.cpp

car rental system | (Global Scope) | operator>>(istream & input, Car & obj)

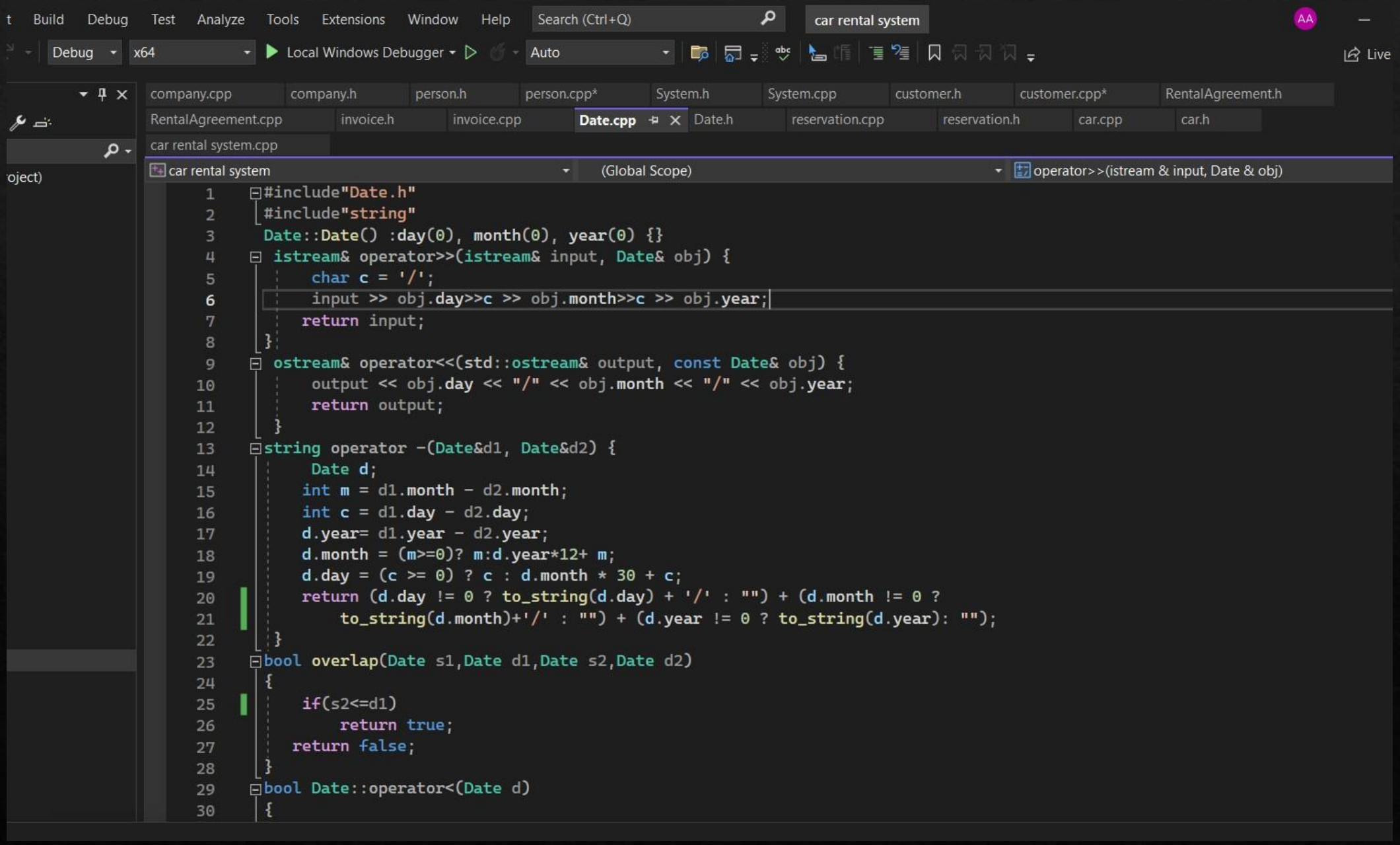
```
58              return false;
59          }
60
61      }availability = 1;
62      return true;
63  }
64  istream& operator>>(istream& input, Car& obj) {
65      cout << "\nEnter car informantion(Id/make/model/year): ";
66      input >>obj.id>> obj.make >> obj.model >> obj.year ;
67      return input;
68  }
69  ostream& operator<<(std::ostream& output, const Car& obj) {
70      cout << "\n\t=====CAR INFORMATION=====" << endl;
71      output <<"Id: "<<obj.id<<" | Make: " << obj.make << " | model : " << obj.model << " | Year : " << obj.year<<endl;
72      if (obj.reserve_count != 0) {
73          for (int i = 0; i < obj.reserve_count; i++) {
74              output << setw(20) << "***Reservation " << i + 1 << " ***"<<endl;
75              output << obj.reserve[i] << endl;
76              output << "Customer id: "<<obj.reserve[i].getCustomerId();
77
78          }
79      }
80      return output;
81  }
82  bool Car::operator<(Car car)
83  {
84      return reserve_count <car.getReserveCount();
85  }
86
87  bool Car::operator==(Car car)
```


Debug
x64
Local Windows Debugger
Auto
company.cpp
company.h
person.h
person.cpp*
System.h
System.cpp
customer.h
customer.cpp*
RentalAgreement.h
RentalAgreement.cpp
invoice.h
invoice.cpp
Date.cpp
Date.h
reservation.cpp
reservation.h
car.cpp
car.h
car rental system.cpp
(1 of 1 project)
car rental system
(Global Scope)
operator>>(istream & input, Car & obj)
79 }
80 return output;
81 }
82 bool Car::operator<(Car car)
83 {
84 return reserve_count <car.getReserveCount();
85 }
86
87 bool Car::operator==(Car car)
88 {
89 return reserve_count ==car.getReserveCount();
90 }
91
92 bool Car::operator>(Car car)
93 {
94 return reserve_count > car.getReserveCount();
95
96 }
97
98 Car::~Car() {
99 delete[]reserve;
100 }
101 Reservation::Reservation(const Reservation& other) {
102 startDate = other.startDate;
103 endDate = other.endDate;
104 c = new Car();
105 *c = *(other.c);
106 }

Header file for class date



Source file for class date



Debug x64 Local Windows Debugger Auto Live Share

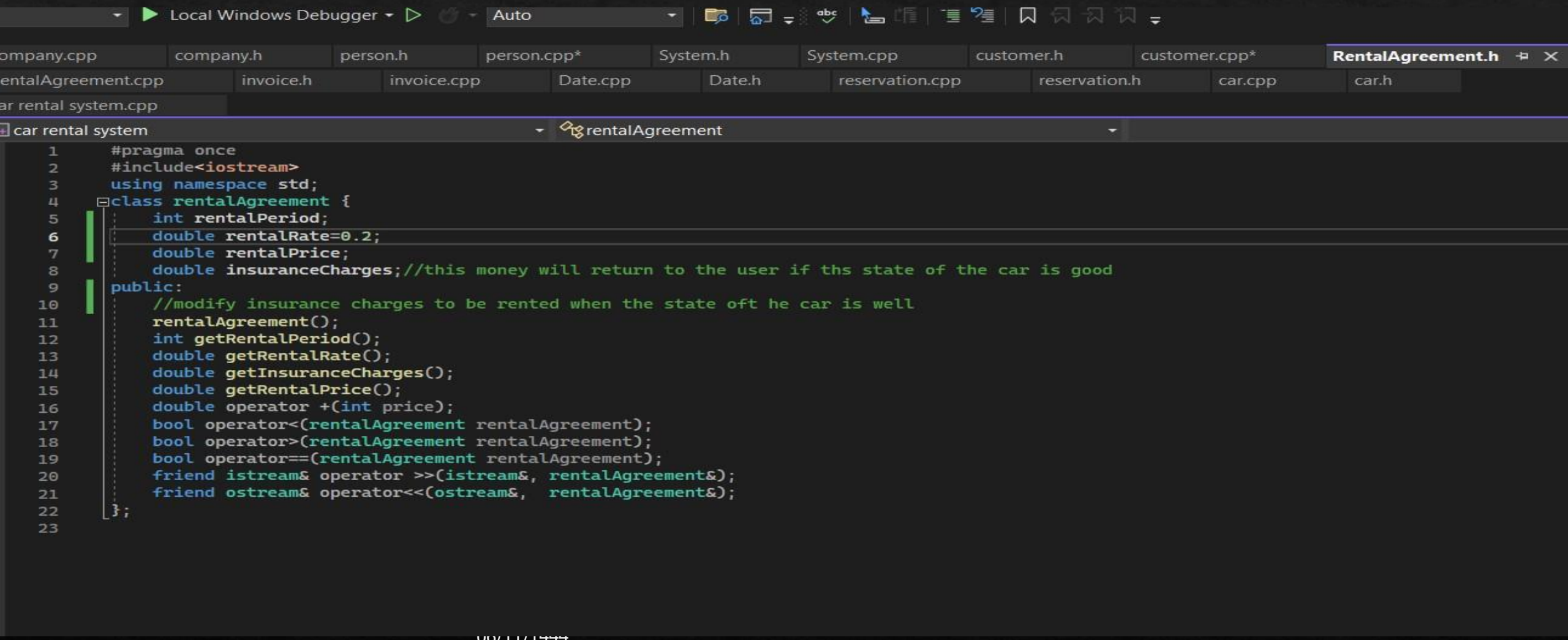
company.cpp | company.h | person.h | person.cpp* | System.h | System.cpp | customer.h | customer.cpp* | RentalAgreement.h
RentalAgreement.cpp | invoice.h | invoice.cpp | **Date.cpp** | Date.h | reservation.cpp | reservation.h | car.cpp | car.h
car rental system.cpp

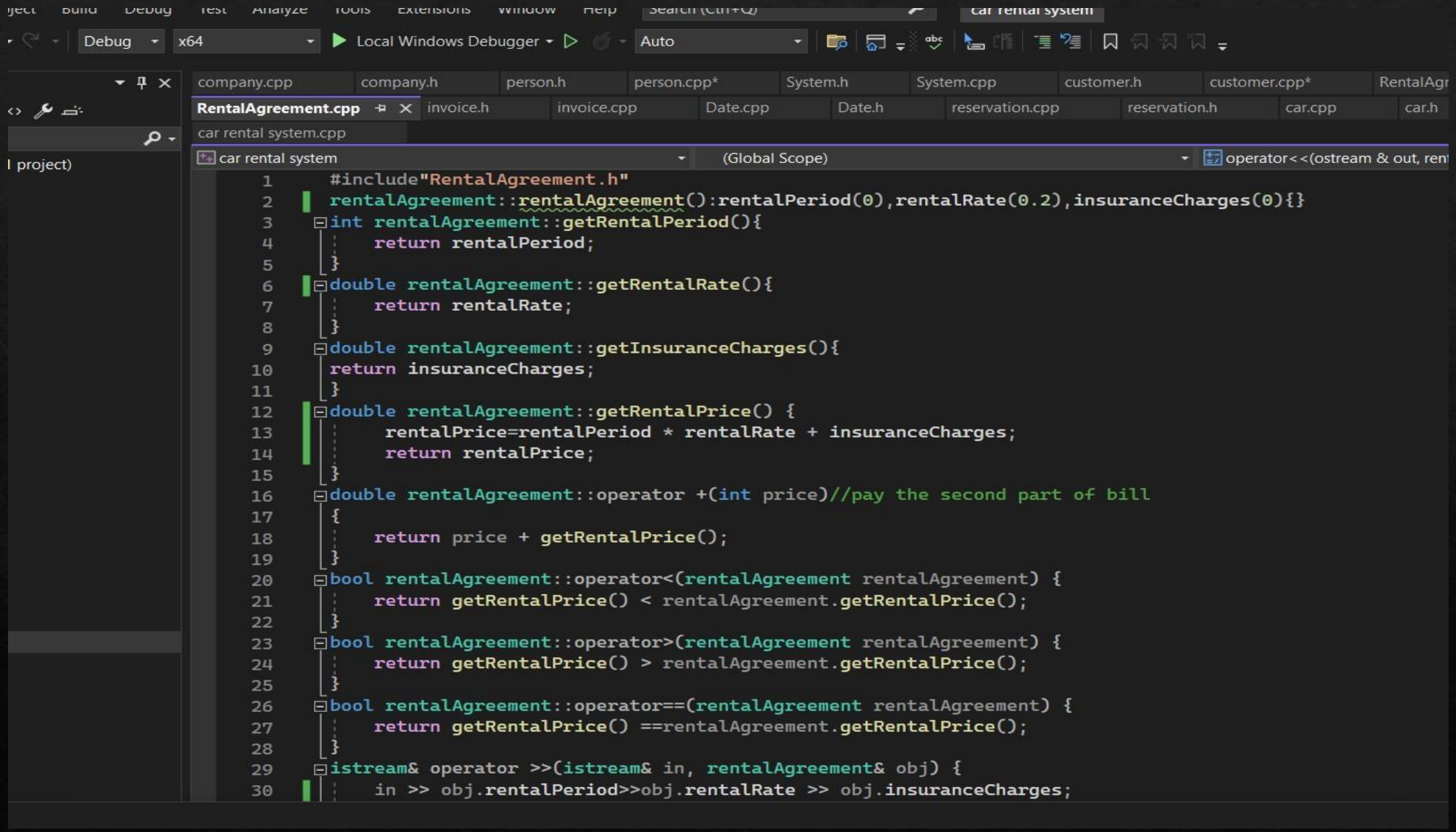
car rental system | (Global Scope) | operator>>(istream & input, Date & obj)

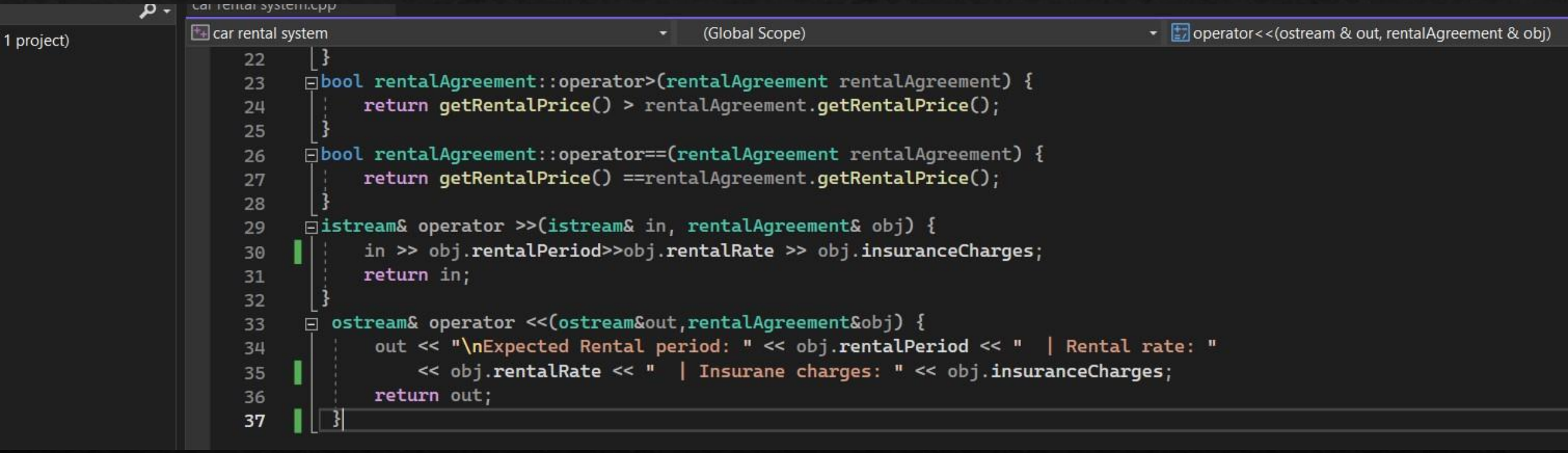
```
28      }
29      bool Date::operator<(Date d)
30      {
31          return(year<d.year)||(year==d.year&&month<d.month)||(year==d.year&&month==d.month&&day<d.day);
32      }
33      bool Date::operator==(Date d)
34      {
35          return year==d.year&&month==d.month&&day==d.day;
36      }
37
38      bool Date::operator>(Date d)
39      {
40          return (year > d.year) || (year == d.year && month > d.month) || (year == d.year && month == d.month && day > d.day);
41      }
42      bool Date::operator>=(Date d) {
43          return (*this == d) || (*this > d);
44      }
45      bool Date::operator<=(Date d) {
46          return (*this == d) || (*this < d);
47      }
```

Toolbox Properties

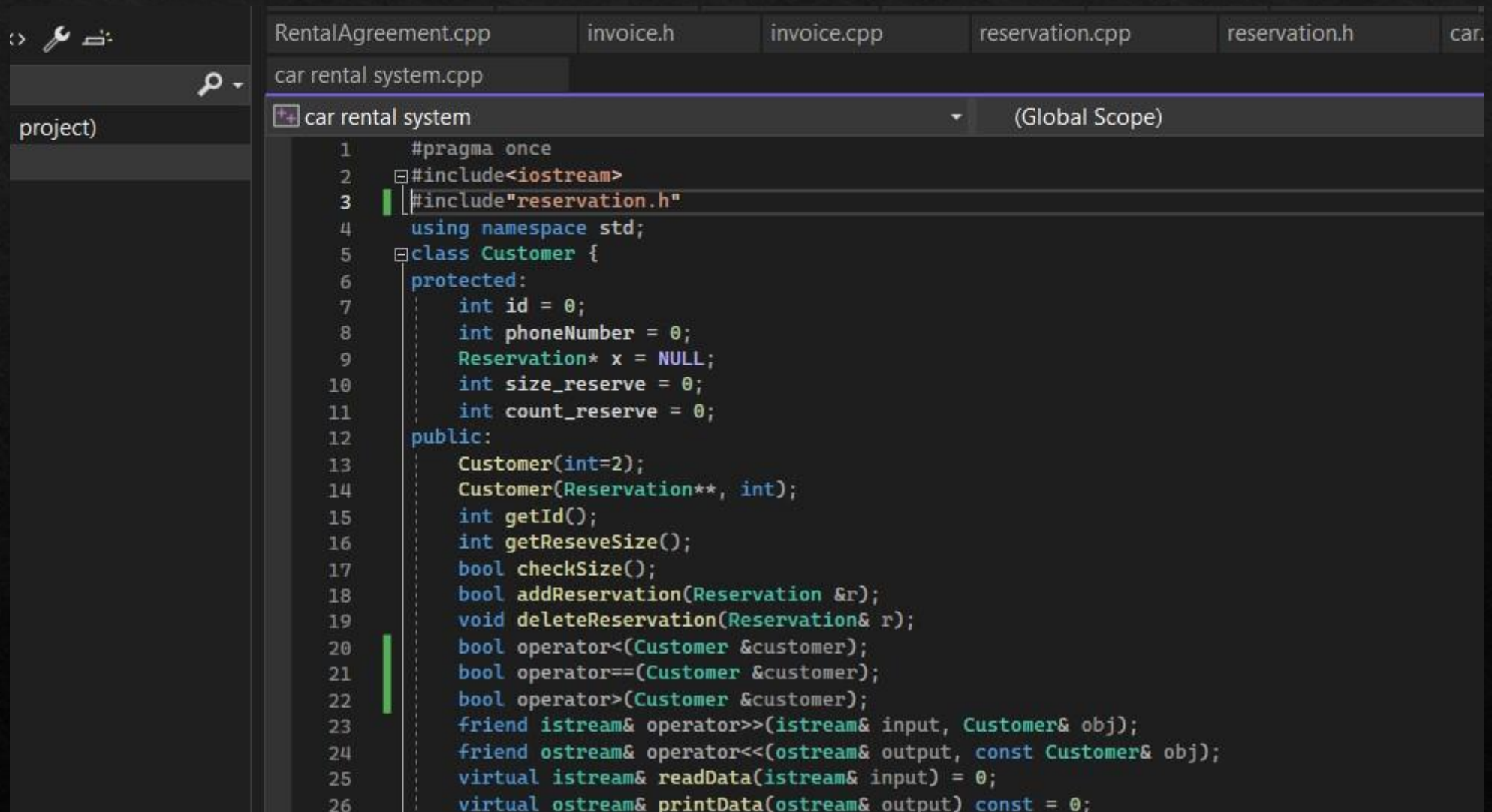
Rental agreement header file




Debug
x64
Local Windows Debugger
Auto
company.cpp
company.h
person.h
person.cpp*
System.h
System.cpp
customer.h
customer.cpp*
RentalAgreement.cpp
invoice.h
invoice.cpp
Date.cpp
Date.h
reservation.cpp
reservation.h
car.cpp
car.h
car rental system.cpp
car rental system
(Global Scope)
#include"RentalAgreement.h"
rentalAgreement::rentalAgreement():rentalPeriod(0),rentalRate(0.2),insuranceCharges(0){}
int rentalAgreement::getRentalPeriod(){
return rentalPeriod;
}
double rentalAgreement::getRentalRate(){
return rentalRate;
}
double rentalAgreement::getInsuranceCharges(){
return insuranceCharges;
}
double rentalAgreement::getRentalPrice() {
rentalPrice=rentalPeriod * rentalRate + insuranceCharges;
return rentalPrice;
}
double rentalAgreement::operator +(int price)//pay the second part of bill
{
return price + getRentalPrice();
}
bool rentalAgreement::operator<(rentalAgreement rentalAgreement) {
return getRentalPrice() < rentalAgreement.getRentalPrice();
}
bool rentalAgreement::operator>(rentalAgreement rentalAgreement) {
return getRentalPrice() > rentalAgreement.getRentalPrice();
}
bool rentalAgreement::operator==(rentalAgreement rentalAgreement) {
return getRentalPrice() ==rentalAgreement.getRentalPrice();
}
istream& operator >>(istream& in, rentalAgreement& obj) {
in >> obj.rentalPeriod>>obj.rentalRate >> obj.insuranceCharges;


1 project)
car rental system
(Global Scope)
operator<<(ostream & out, rentalAgreement & obj)
22 }
23 bool rentalAgreement::operator>(rentalAgreement rentalAgreement) {
24 return getRentalPrice() > rentalAgreement.getRentalPrice();
25 }
26 bool rentalAgreement::operator==(rentalAgreement rentalAgreement) {
27 return getRentalPrice() ==rentalAgreement.getRentalPrice();
28 }
29 istream& operator >>(istream& in, rentalAgreement& obj) {
30 in >> obj.rentalPeriod>>obj.rentalRate >> obj.insuranceCharges;
31 return in;
32 }
33 ostream& operator <<(ostream&out,rentalAgreement&obj) {
34 out << "\nExpected Rental period: " << obj.rentalPeriod << " | Rental rate: "
35 << obj.rentalRate << " | Insurane charges: " << obj.insuranceCharges;
36 return out;
37 }

Header file for class customer

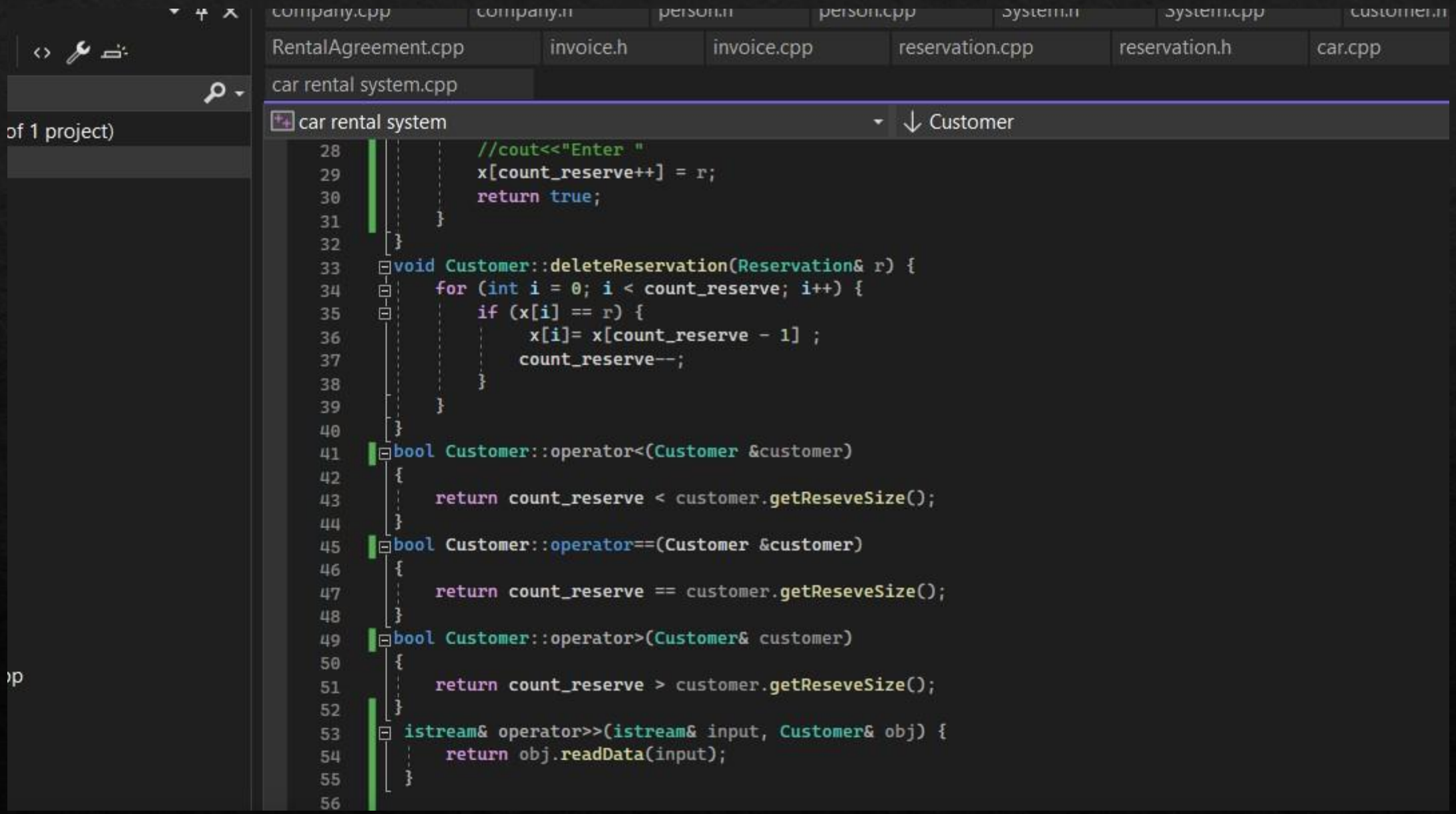


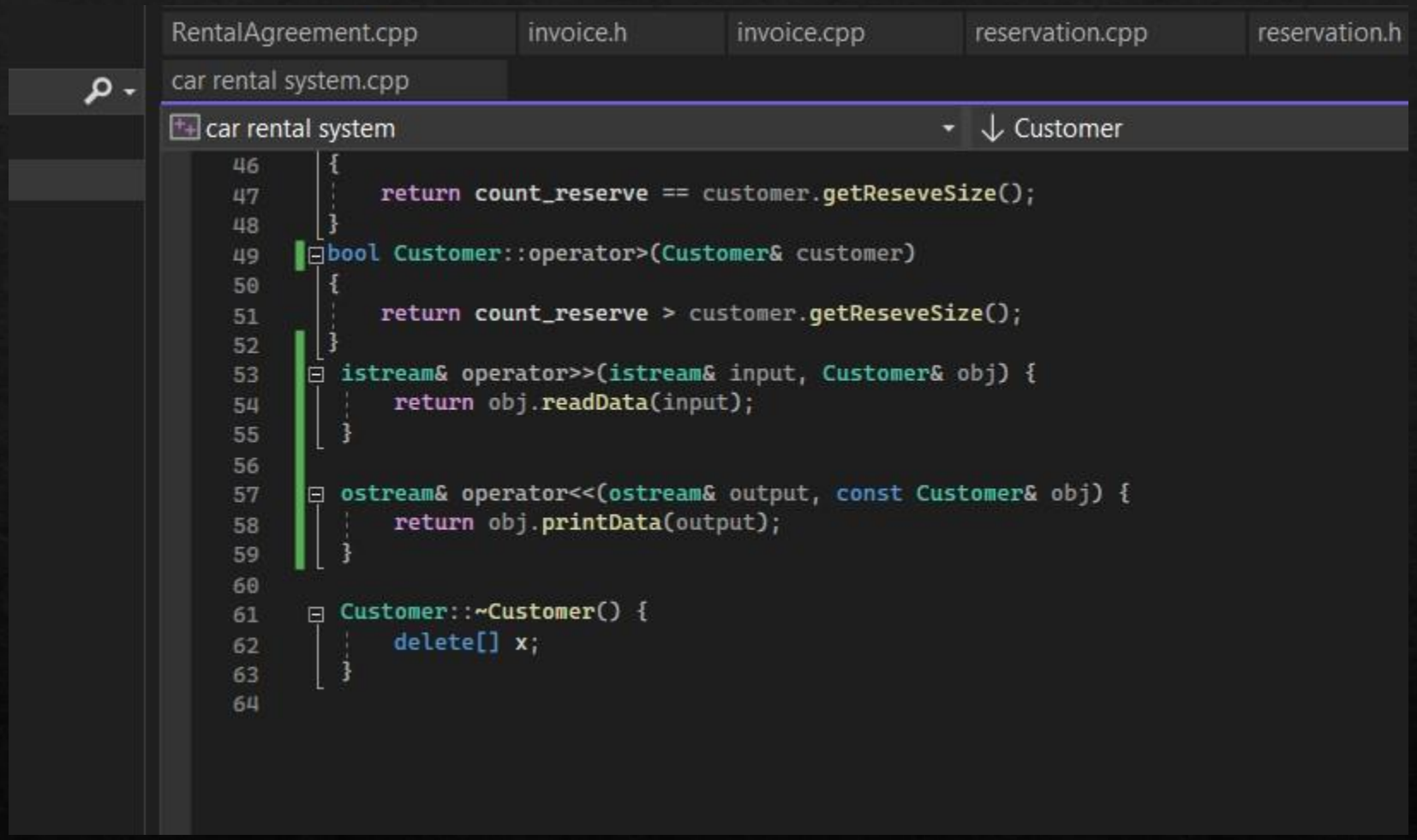
Source file for customer

ntal system.cpp

r rental system ▾ | ↓ Customer ▾

```cpp
#include"customer.h"
Customer::Customer(int c){
    size_reserve = c;
    x = new Reservation[size_reserve];
    }
Customer::Customer(Reservation**reserveArr, int s) {
    count_reserve=size_reserve = s;
    x = new Reservation[size_reserve];
    for (int i = 0; i < size_reserve; i++) {
        x[i] = Reservation(*reserveArr[i]);
    }
}
int Customer::getId() {
    return id;
}
int Customer::getReseveSize() {
    return size_reserve;
}
bool Customer::checkSize() {
    return count_reserve <= size_reserve;
}
bool Customer::addReservation(Reservation &r) {
    if (count_reserve == size_reserve) {
        cout << "You exceeded the utmost number of reservations..You can't add any more";
        return false;
    }
    else {
        //cout<<"Enter "
        x[count_reserve++] = r;
```


company.cpp
company.h
person.h
person.cpp
System.h
System.cpp
customer.h
RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cpp
car rental system.cpp
car rental system
Customer
of 1 project)
//cout<<"Enter "
x[count_reserve++] = r;
return true;
}
}
void Customer::deleteReservation(Reservation& r) {
for (int i = 0; i < count_reserve; i++) {
if (x[i] == r) {
x[i]= x[count_reserve - 1] ;
count_reserve--;
}
}
}
bool Customer::operator<(Customer &customer)
{
return count_reserve < customer.getReseveSize();
}
bool Customer::operator==(Customer &customer)
{
return count_reserve == customer.getReseveSize();
}
bool Customer::operator>(Customer& customer)
{
return count_reserve > customer.getReseveSize();
}
istream& operator>>(istream& input, Customer& obj) {
return obj.readData(input);
}


RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car rental system.cpp
car rental system
Customer
{
return count_reserve == customer.getReseveSize();
}
bool Customer::operator>(Customer& customer)
{
return count_reserve > customer.getReseveSize();
}
istream& operator>>(istream& input, Customer& obj) {
return obj.readData(input);
}
ostream& operator<<(ostream& output, const Customer& obj) {
return obj.printData(output);
}
Customer::~Customer() {
delete[] x;
}

source file for person

car rental system.cpp

car rental system | Person | readData(istream & input)

```cpp
#include"person.h"
#include<iomanip>
#include<string>
istream& Person ::readData(istream& input)  {
    cout << "Enter customer's details (id, phone number, full name, address): ";
    input >> id >> phoneNumber >> full_name >> address;
    return input;
}
ostream& Person::printData(ostream& output) const  {
    output << "Customer's details: " << endl << "id: " << id << " | Phone number: " << phoneNumber << endl;
    output << "Full name: " << full_name << " | Address: " << address << endl;
    return output;
}
```

Header file for person

car rental system.cpp

car rental system — Person

```
1   #pragma once
2   #include"customer.h"
3   class Person :public Customer {
4       string full_name;
5       string address;
6   public:
7       istream& readData(istream& input)override;
8       ostream& printData(ostream& output) const override;
9   };
10
```

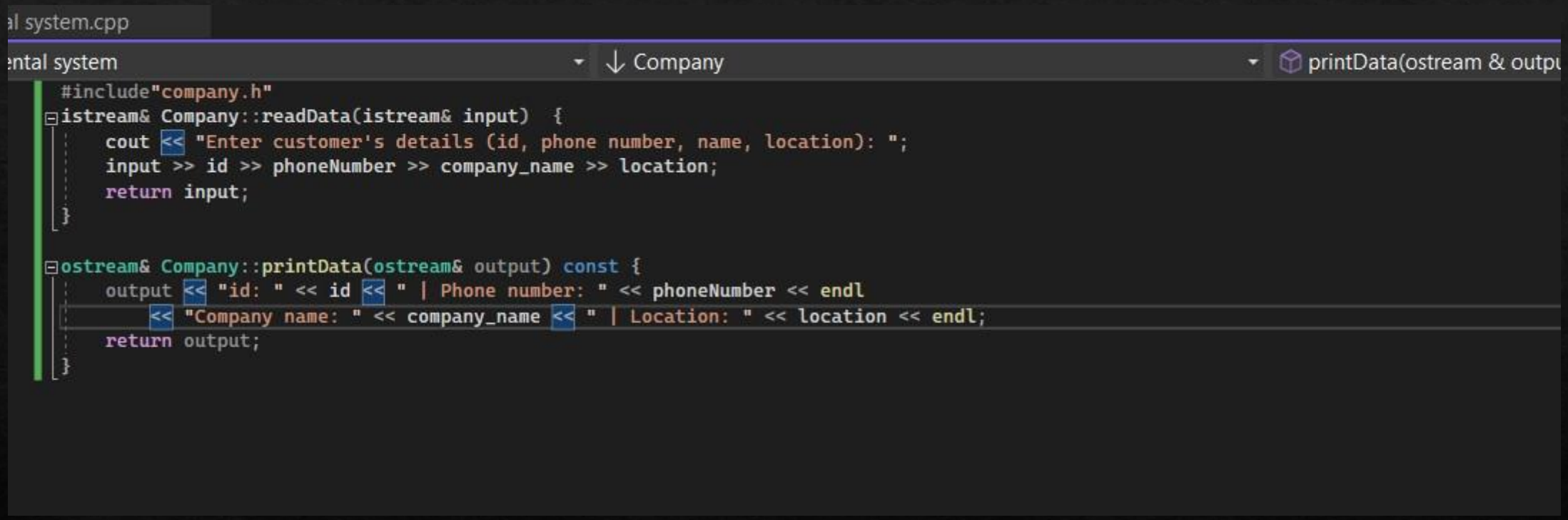
Company header file

car rental system.cpp

car rental system | Company

```
1   #pragma once
2   #include"customer.h"
3   class Company :public Customer {
4       string location;
5       string company_name;
6   public:
7       istream& readData(istream& input)override;
8       ostream& printData(ostream& output) const override;
9   };
10
```

Company source file



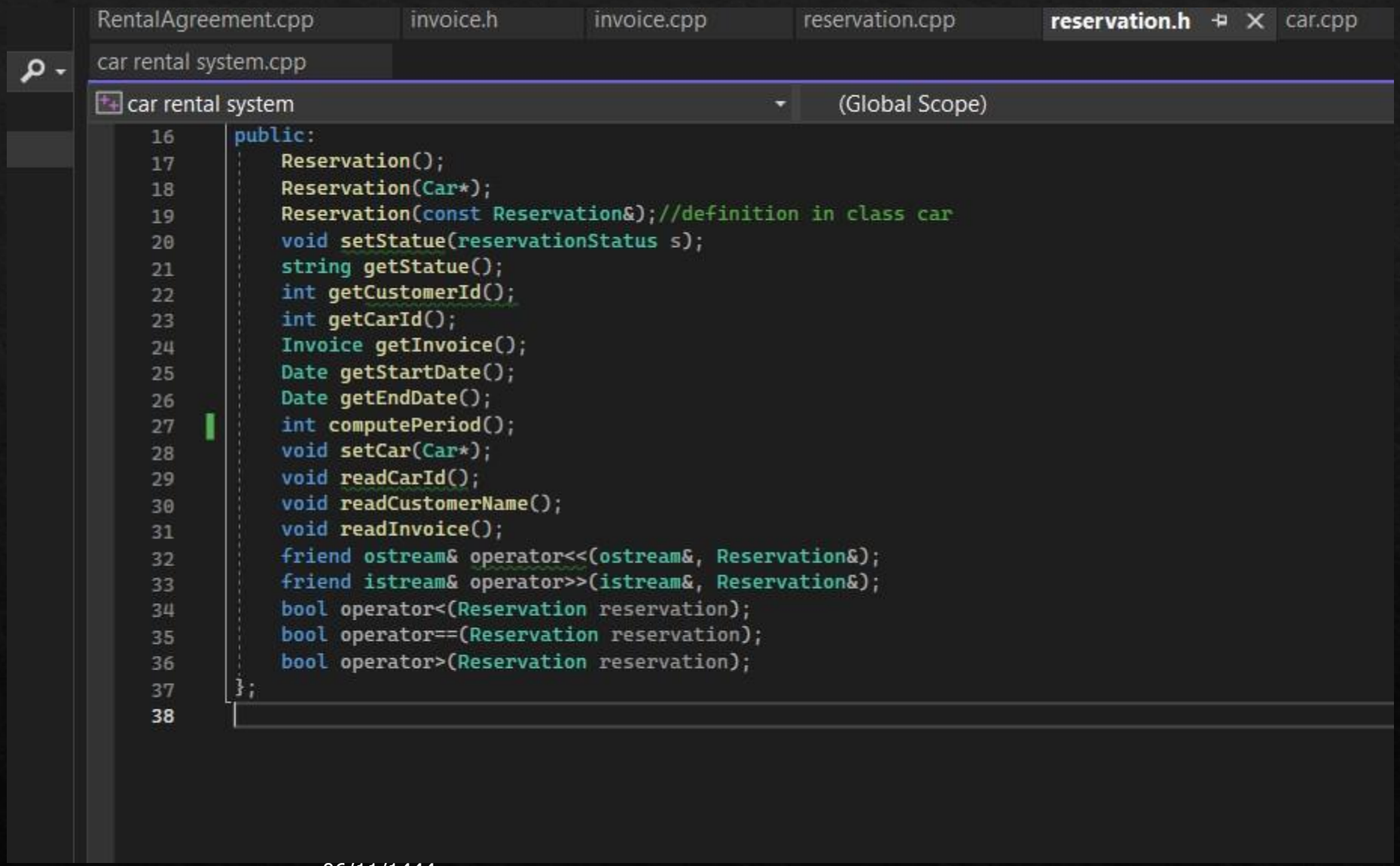
Reservation header file

RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | **reservation.h** | car.cpp

car rental system.cpp

car rental system | (Global Scope)

```
1   #pragma once
2   #include"Date.h"
3   #include"invoice.h"
4   class Car;
5   class Customer;
6   enum reservationStatus { notCompleted, completed, canceled, still };
7   class Reservation {
8       Date startDate;
9       Date endDate;
10      Car* c=NULL;
11      Customer* m_customer=NULL;
12      int customerId=0;
13      int carId=0;
14      Invoice invoice;
15      string status;
16  public:
17      Reservation();
18      Reservation(Car*);
19      Reservation(const Reservation&);//definition in class car
20      void setStatue(reservationStatus s);
21      string getStatue();
22      int getCustomerId();
23      int getCarId();
24      Invoice getInvoice();
25      Date getStartDate();
26      Date getEndDate();
27      int computePeriod();
28      void setCar(Car*);
29      void readCarId();
```


RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cpp
car rental system.cpp
car rental system
(Global Scope)
public:
Reservation();
Reservation(Car*);
Reservation(const Reservation&);//definition in class car
void setStatue(reservationStatus s);
string getStatue();
int getCustomerId();
int getCarId();
Invoice getInvoice();
Date getStartDate();
Date getEndDate();
int computePeriod();
void setCar(Car*);
void readCarId();
void readCustomerName();
void readInvoice();
friend ostream& operator<<(ostream&, Reservation&);
friend istream& operator>>(istream&, Reservation&);
bool operator<(Reservation reservation);
bool operator==(Reservation reservation);
bool operator>(Reservation reservation);
};

car rental system.cpp

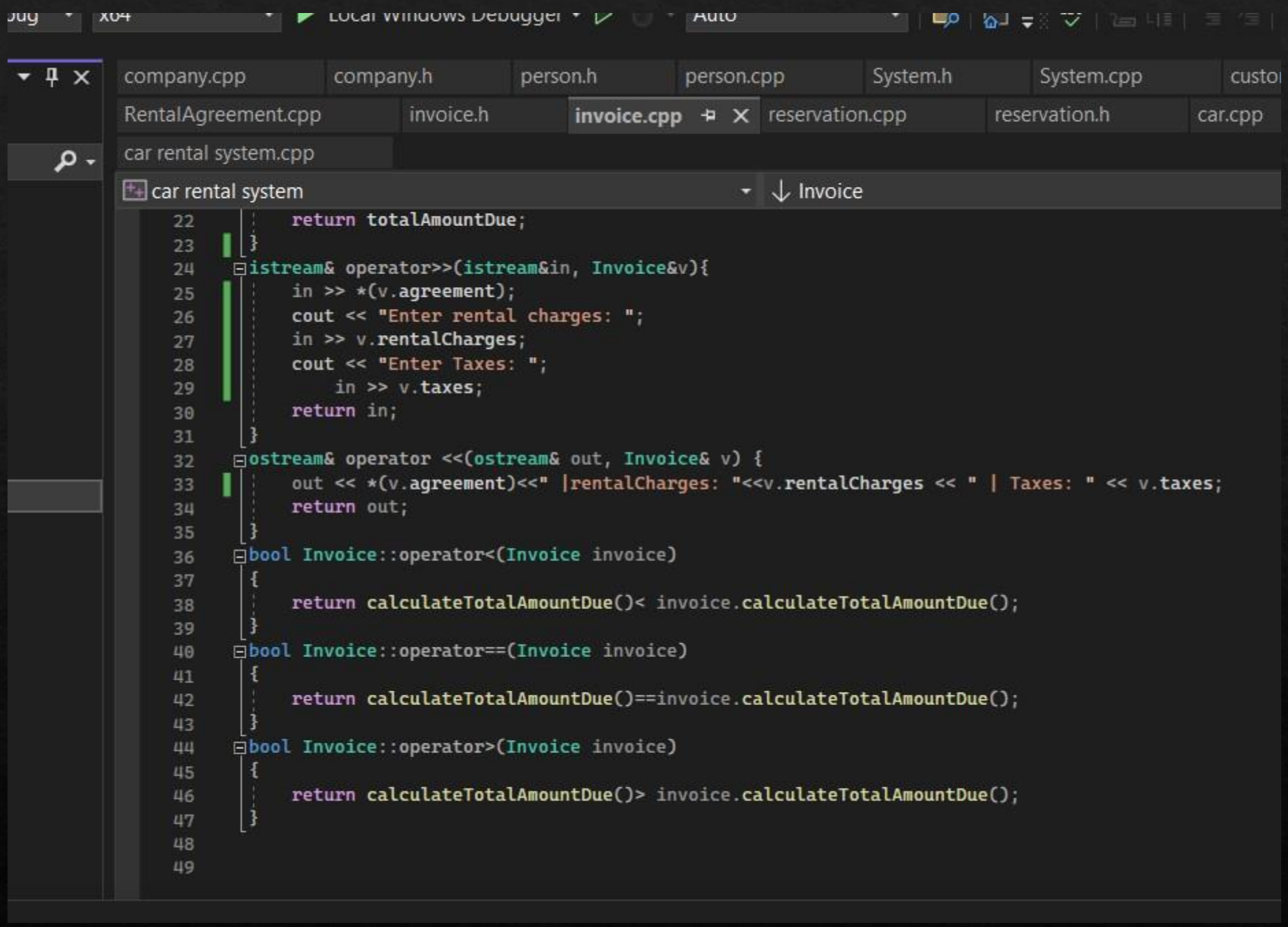
car rental system | Invoice | calculat

```
1   #pragma once
2   #include"RentalAgreement.h"
3   class Invoice {
4       double rentalCharges=0;
5       double taxes=0;
6       double totalAmountDue=0;
7       rentalAgreement* agreement=NULL;
8   public:
9       Invoice();
10      Invoice(rentalAgreement*);
11      void setAgreement(rentalAgreement*);
12      void readtaxes();
13      double getRentalCharges();
14      double calculateTotalAmountDue();
15      friend istream& operator>>(istream&, Invoice&);
16      friend ostream& operator <<(ostream& out, Invoice& obj);
17      //int operator -(int total_amount);//what i should pay the next time
18      bool operator<(Invoice invoice);
19      bool operator==(Invoice invoice);
20      bool operator>(Invoice invoice);
21      //ostream& operator <<(ostream& output);
22  };
23
```

car rental system.cpp

car rental system ↓ Invoice

```cpp
#include"invoice.h"
Invoice::Invoice():taxes(0), rentalCharges(0), totalAmountDue(0) {
    agreement = new rentalAgreement;
}
Invoice::Invoice(rentalAgreement*p){
    agreement = new rentalAgreement;
    agreement = p;
}
void Invoice::setAgreement(rentalAgreement* p) {
    agreement = new rentalAgreement;
    agreement = p;
}
void Invoice::readtaxes() {
    cout << "\nEnter Taxes: ";
    cin >> taxes;
}
double Invoice::getRentalCharges() {
    return rentalCharges;
}
double Invoice::calculateTotalAmountDue() {
    totalAmountDue = rentalCharges + agreement->getRentalPrice()+taxes;
    return totalAmountDue;
}
istream& operator>>(istream&in, Invoice&v){
    in >> *(v.agreement);
    cout << "Enter rental charges: ";
    in >> v.rentalCharges;
    cout << "Enter Taxes: ";
        in >> v.taxes;
```

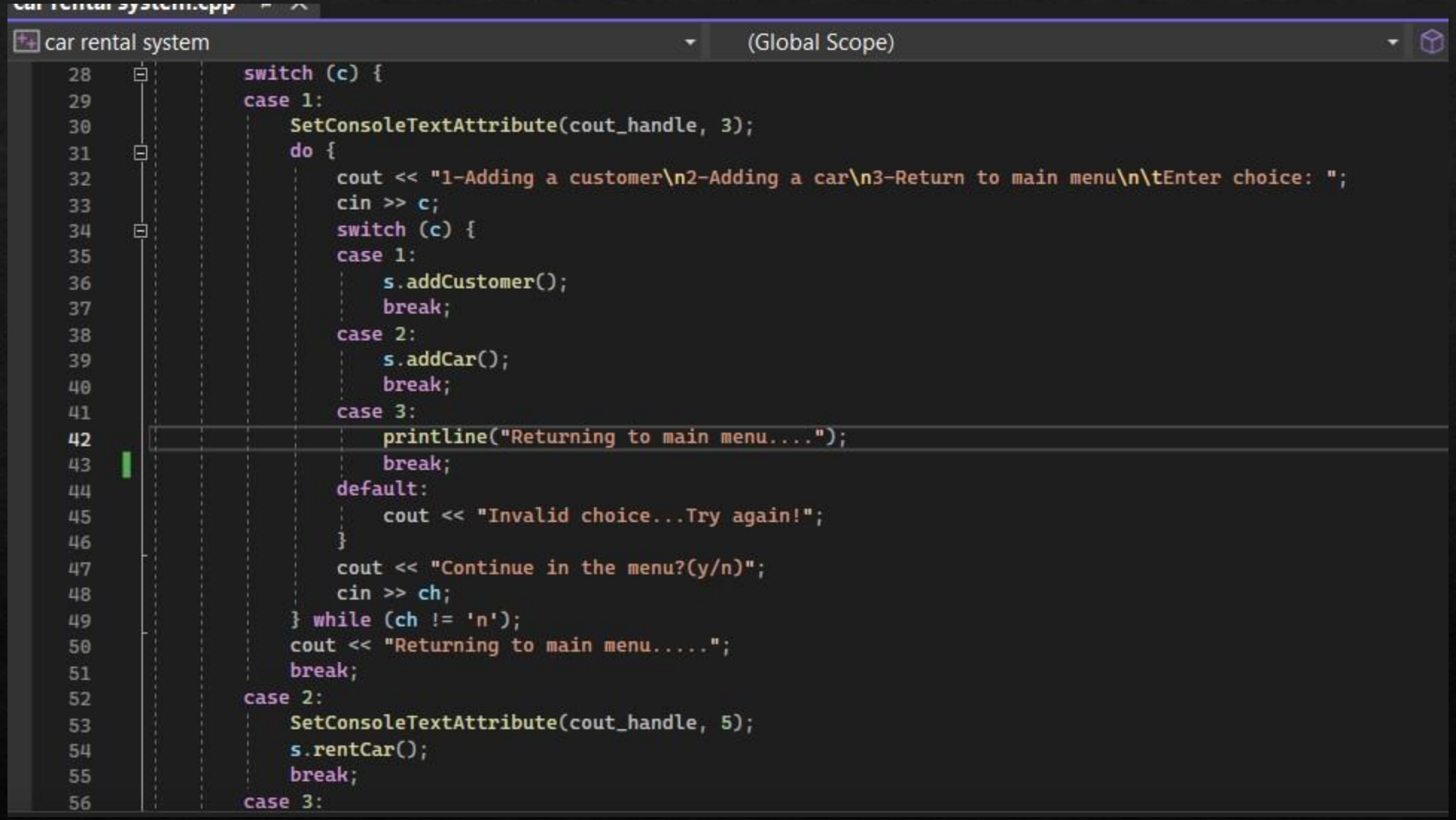

company.cpp
company.h
person.h
person.cpp
System.h
System.cpp
RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cpp
car rental system.cpp
car rental system
Invoice
return totalAmountDue;
}
istream& operator>>(istream&in, Invoice&v){
in >> *(v.agreement);
cout << "Enter rental charges: ";
in >> v.rentalCharges;
cout << "Enter Taxes: ";
in >> v.taxes;
return in;
}
ostream& operator <<(ostream& out, Invoice& v) {
out << *(v.agreement)<<" |rentalCharges: "<<v.rentalCharges << " | Taxes: " << v.taxes;
return out;
}
bool Invoice::operator<(Invoice invoice)
{
return calculateTotalAmountDue()< invoice.calculateTotalAmountDue();
}
bool Invoice::operator==(Invoice invoice)
{
return calculateTotalAmountDue()==invoice.calculateTotalAmountDue();
}
bool Invoice::operator>(Invoice invoice)
{
return calculateTotalAmountDue()> invoice.calculateTotalAmountDue();
}

RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | reservation.h | car.cpp | car.h | Date.h | Date.cpp

car rental system.cpp

car rental system | (Global Scope) | main()

```cpp

#include <iostream>
#include<cstdlib>
#include<iomanip>
#include<windows.h>
#include<string>
#include"System.h"
#include"person.h"
#include"company.h"
using namespace std;
void printline(string s, bool l = true);//defined in system .cpp
HANDLE cout_handle = GetStdHandle(STD_OUTPUT_HANDLE);
int main()
{
    SetConsoleTextAttribute(cout_handle, 6);
    System s(3,3);
    char ch;
    int c = -1;
    while (c != 0) {
        SetConsoleTextAttribute(cout_handle, 6);
        printline("\t\t\t===CAR RENTAL SYSTEM===");
        printline("where do you want to go :\n\t1.Adding cars and customers");
        printline("\t2.Rent car system\n\t3.Return a car\n\t4.View available cars");
        printline("\t5.View rental history\n\t6.Generate Reports");
        printline("Enter selection: ",false);
        cin >> c;
        system("cls");
        switch (c) {
        case 1:
```


car rental system
(Global Scope)
switch (c) {
case 1:
SetConsoleTextAttribute(cout_handle, 3);
do {
cout << "1-Adding a customer\n2-Adding a car\n3-Return to main menu\n\tEnter choice: ";
cin >> c;
switch (c) {
case 1:
s.addCustomer();
break;
case 2:
s.addCar();
break;
case 3:
printline("Returning to main menu....");
break;
default:
cout << "Invalid choice...Try again!";
}
cout << "Continue in the menu?(y/n)";
cin >> ch;
} while (ch != 'n');
cout << "Returning to main menu.....";
break;
case 2:
SetConsoleTextAttribute(cout_handle, 5);
s.rentCar();
break;
case 3:

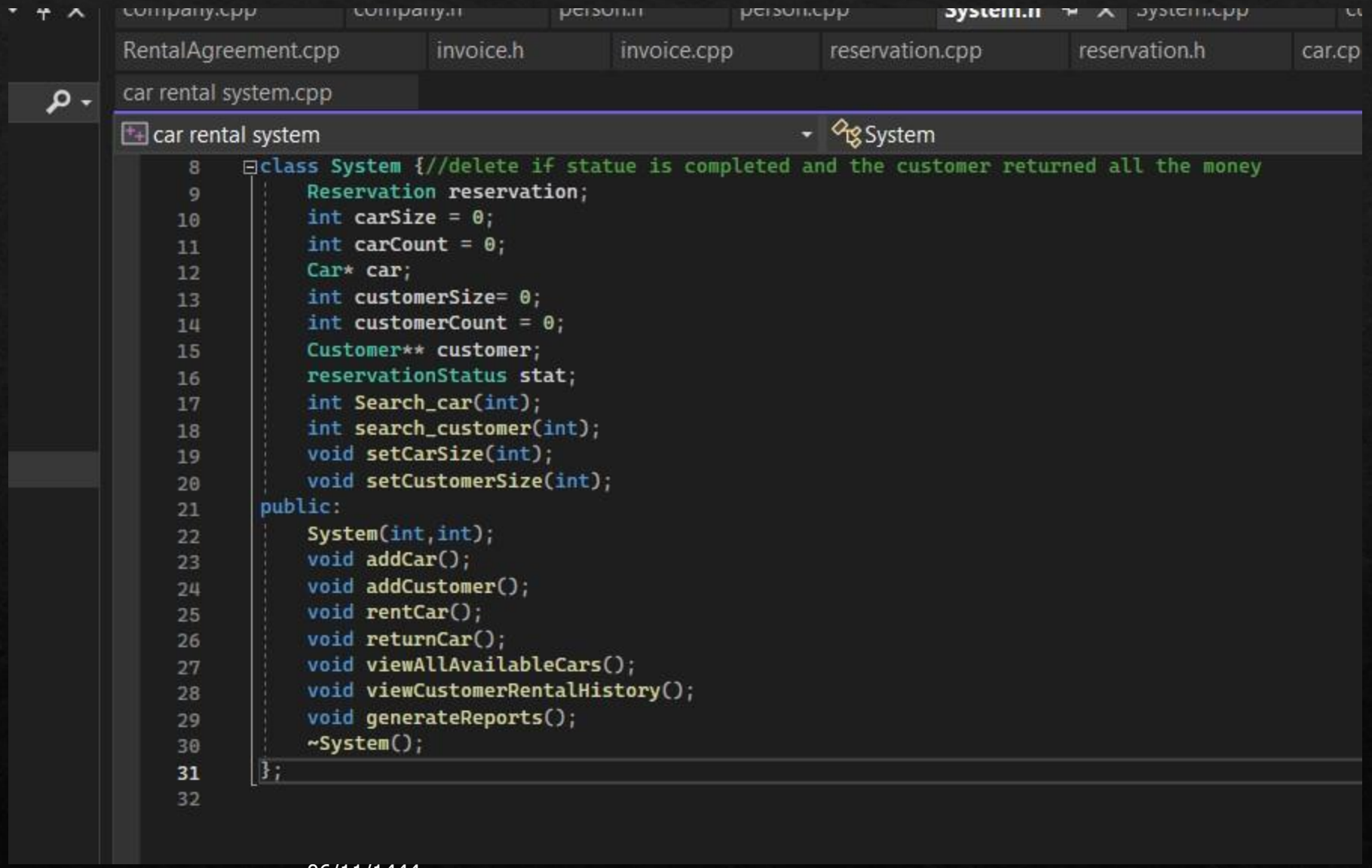
company.cpp | company.h | person.h | person.cpp | System.h | System.cpp

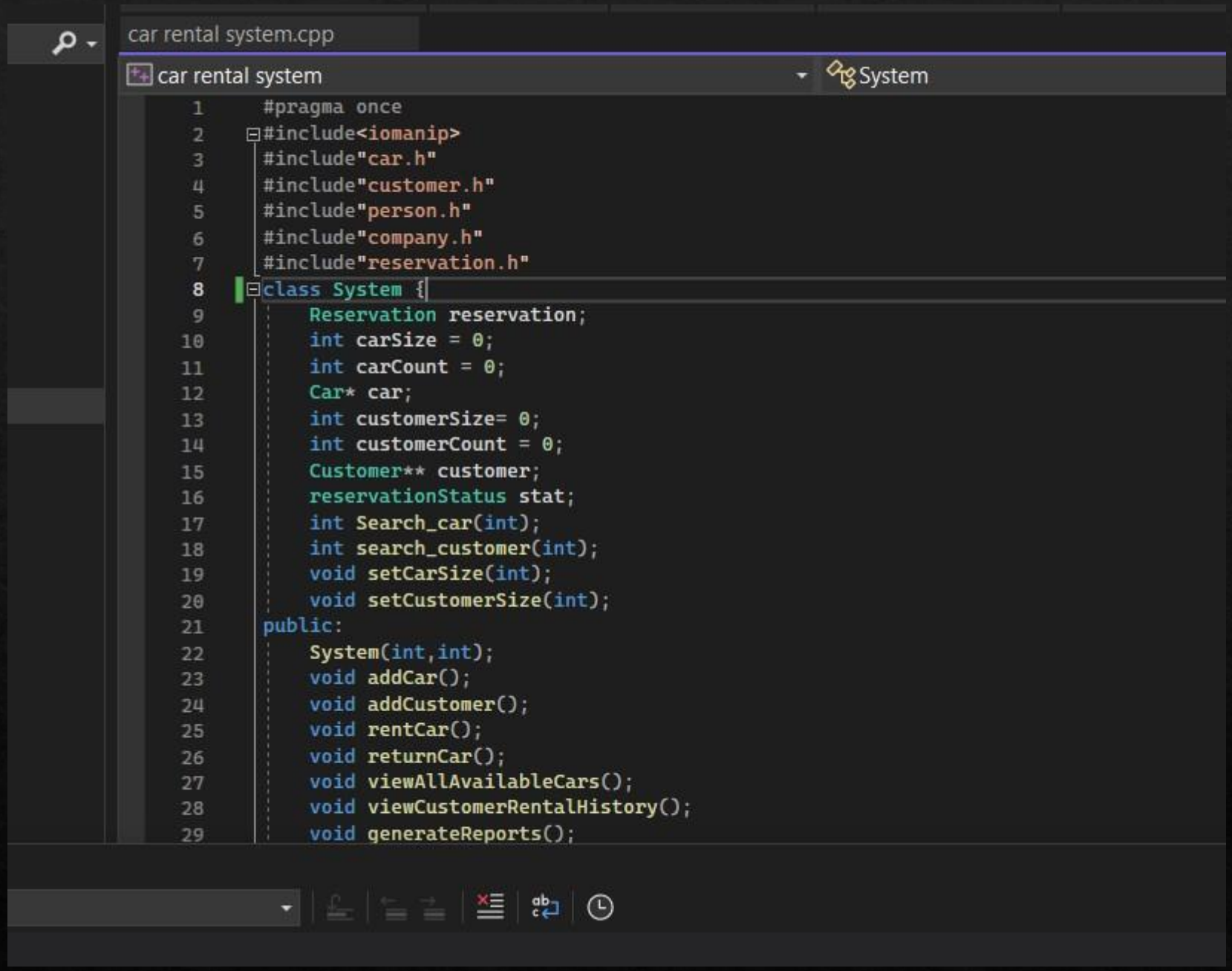
RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | reservation.h | car.cp

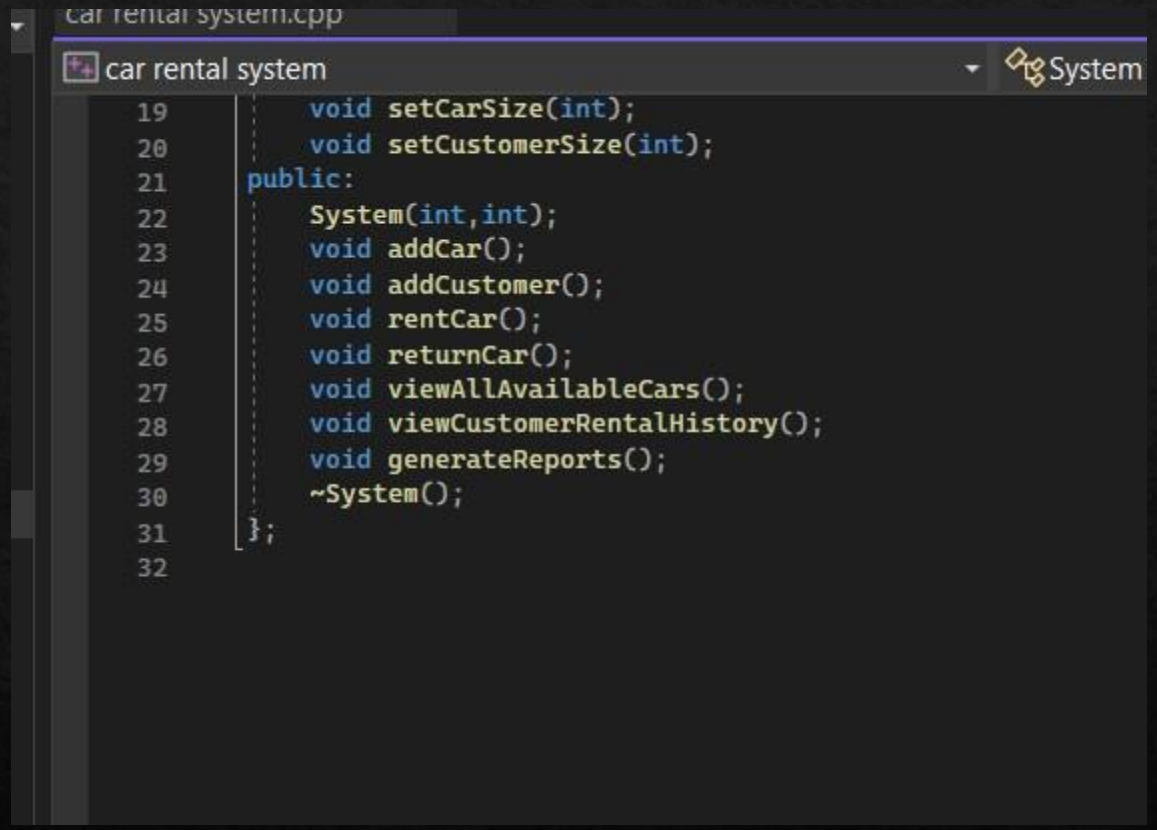
car rental system.cpp

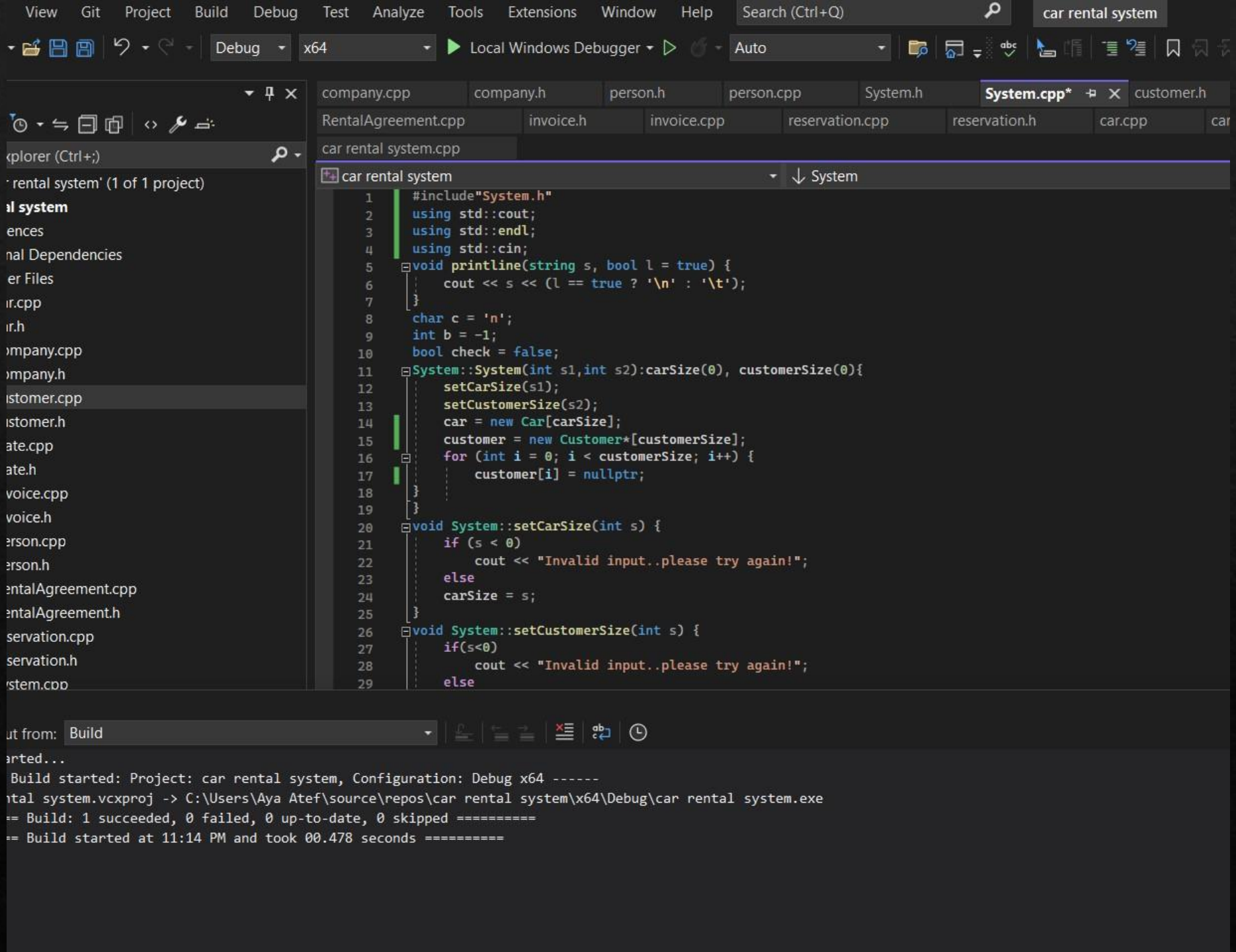
car rental system | (Global Scope)

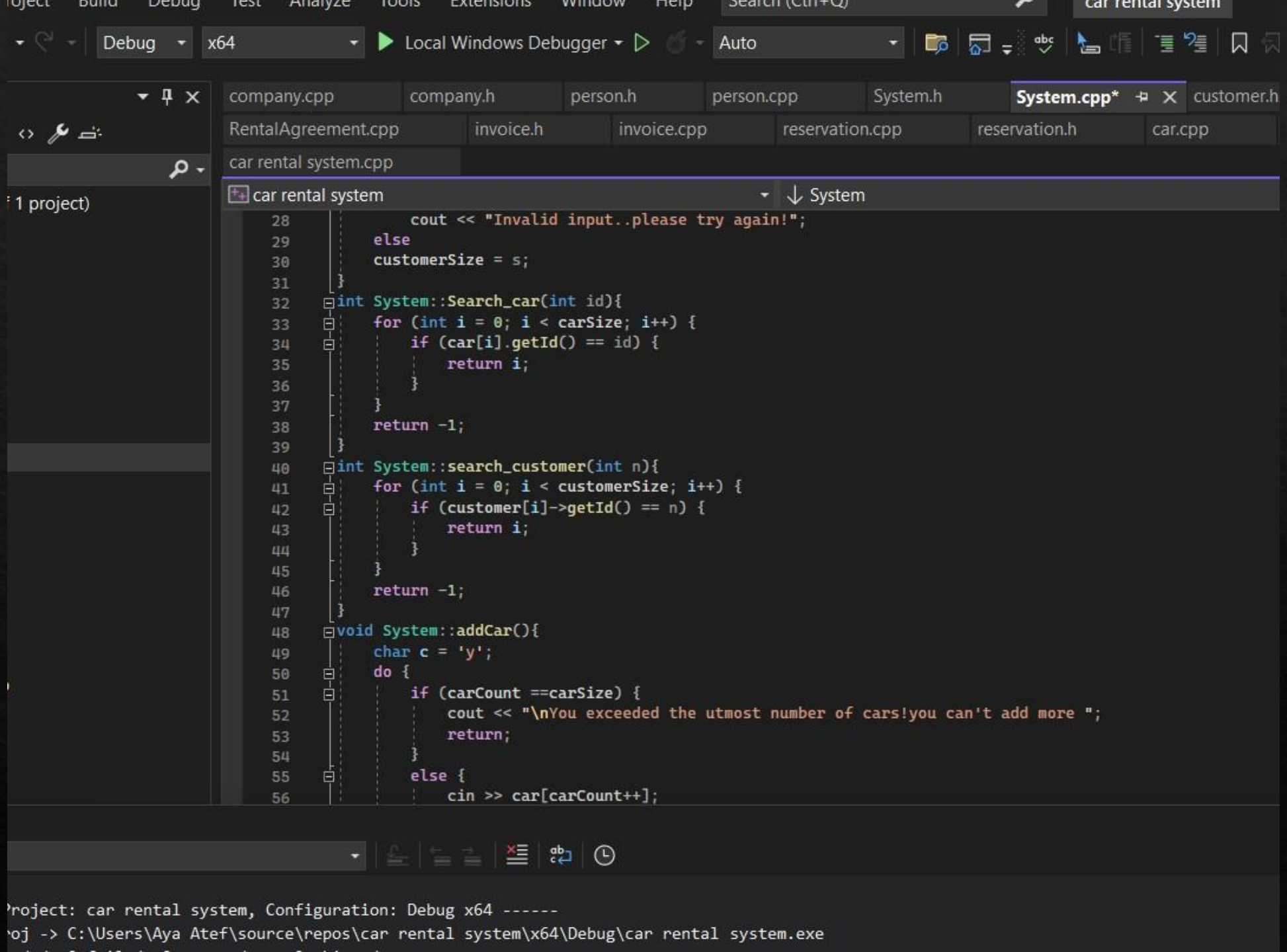
```
55              break;
56          case 3:
57              SetConsoleTextAttribute(cout_handle, 4);
58              s.returnCar();
59              break;
60          case 4:
61              SetConsoleTextAttribute(cout_handle, 3);
62              s.viewAllAvailableCars();
63              break;
64          case 5:
65              SetConsoleTextAttribute(cout_handle, 9);
66              s.viewCustomerRentalHistory();
67              break;
68          case 6:
69              SetConsoleTextAttribute(cout_handle, 11);
70              s.generateReports();
71              break;
72          case 0:
73              printline("\t\tProgram finished.....Nice time with you:)");
74              return 0;
75          default:
76              printline("Wrong choice...Try again!");
77          }
78          Sleep(3000);
79          system("cls");
80      }
81  }
82
```

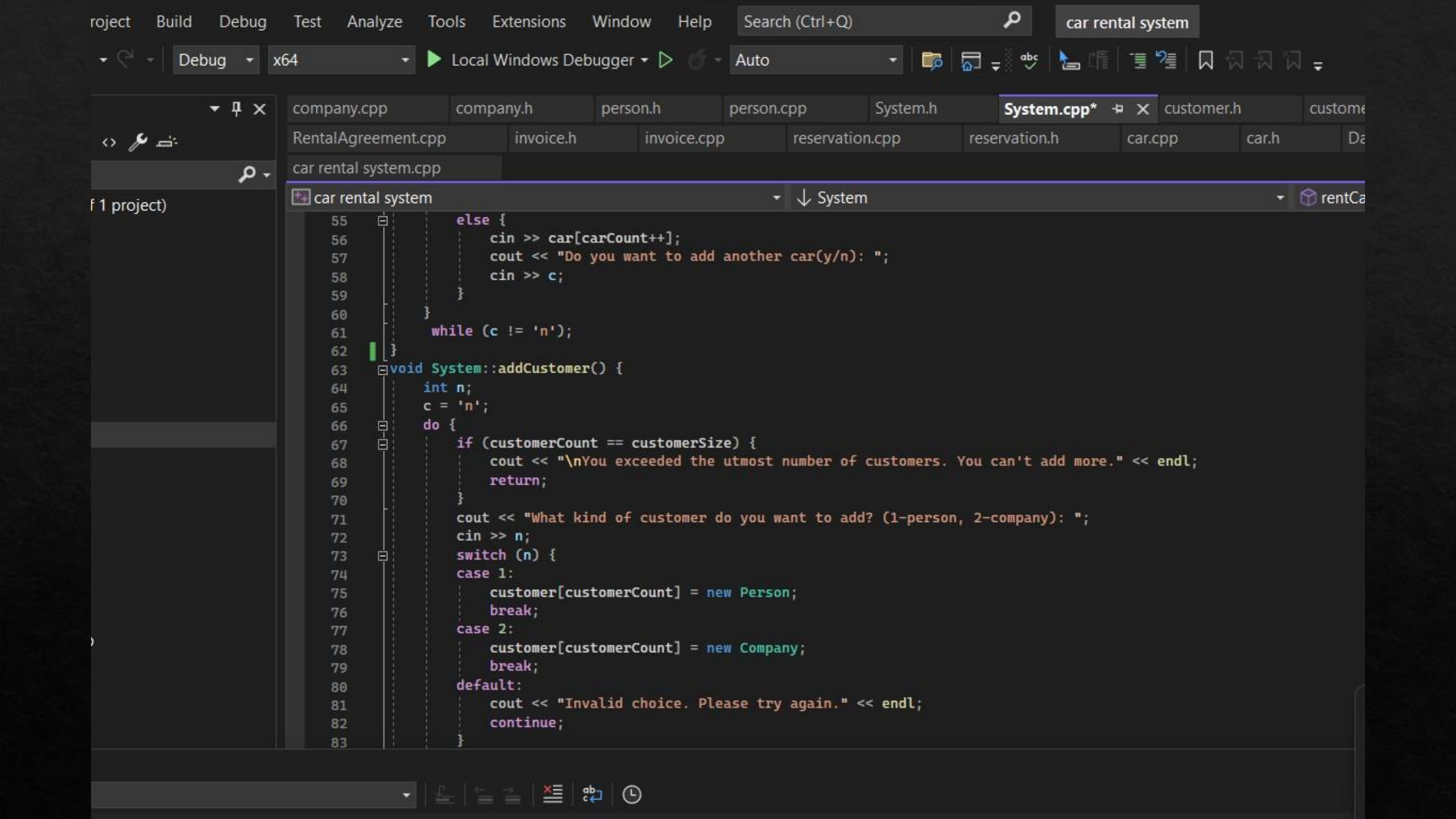

RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cp
car rental system.cpp
car rental system
System
class System {//delete if statue is completed and the customer returned all the money
Reservation reservation;
int carSize = 0;
int carCount = 0;
Car* car;
int customerSize= 0;
int customerCount = 0;
Customer** customer;
reservationStatus stat;
int Search_car(int);
int search_customer(int);
void setCarSize(int);
void setCustomerSize(int);
public:
System(int,int);
void addCar();
void addCustomer();
void rentCar();
void returnCar();
void viewAllAvailableCars();
void viewCustomerRentalHistory();
void generateReports();
~System();
};


car rental system.cpp
car rental system
System
#pragma once
#include<iomanip>
#include"car.h"
#include"customer.h"
#include"person.h"
#include"company.h"
#include"reservation.h"
class System {
Reservation reservation;
int carSize = 0;
int carCount = 0;
Car* car;
int customerSize= 0;
int customerCount = 0;
Customer** customer;
reservationStatus stat;
int Search_car(int);
int search_customer(int);
void setCarSize(int);
void setCustomerSize(int);
public:
System(int,int);
void addCar();
void addCustomer();
void rentCar();
void returnCar();
void viewAllAvailableCars();
void viewCustomerRentalHistory();
void generateReports();


car rental system.cpp
car rental system
System
19 void setCarSize(int);
20 void setCustomerSize(int);
21 public:
22 System(int,int);
23 void addCar();
24 void addCustomer();
25 void rentCar();
26 void returnCar();
27 void viewAllAvailableCars();
28 void viewCustomerRentalHistory();
29 void generateReports();
30 ~System();
31 };
32


View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
car rental system
Debug x64 Local Windows Debugger Auto
company.cpp company.h person.h person.cpp System.h System.cpp* customer.h
RentalAgreement.cpp invoice.h invoice.cpp reservation.cpp reservation.h car.cpp
car rental system.cpp
car rental system
System
#include"System.h"
using std::cout;
using std::endl;
using std::cin;
void printline(string s, bool l = true) {
cout << s << (l == true ? '\n' : '\t');
}
char c = 'n';
int b = -1;
bool check = false;
System::System(int s1,int s2):carSize(0), customerSize(0){
setCarSize(s1);
setCustomerSize(s2);
car = new Car[carSize];
customer = new Customer*[customerSize];
for (int i = 0; i < customerSize; i++) {
customer[i] = nullptr;
}
}
void System::setCarSize(int s) {
if (s < 0)
cout << "Invalid input..please try again!";
else
carSize = s;
}
void System::setCustomerSize(int s) {
if(s<0)
cout << "Invalid input..please try again!";
else
Build started: Project: car rental system, Configuration: Debug x64 ------
tal system.vcxproj -> C:\Users\Aya Atef\source\repos\car rental system\x64\Debug\car rental system.exe
== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
== Build started at 11:14 PM and took 00.478 seconds ==========


Debug
x64
Local Windows Debugger
Auto
company.cpp
company.h
person.h
person.cpp
System.h
System.cpp*
customer.h
RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cpp
car rental system.cpp
car rental system
System
1 project)
cout << "Invalid input..please try again!";
else
customerSize = s;
}
int System::Search_car(int id){
for (int i = 0; i < carSize; i++) {
if (car[i].getId() == id) {
return i;
}
}
return -1;
}
int System::search_customer(int n){
for (int i = 0; i < customerSize; i++) {
if (customer[i]->getId() == n) {
return i;
}
}
return -1;
}
void System::addCar(){
char c = 'y';
do {
if (carCount ==carSize) {
cout << "\nYou exceeded the utmost number of cars!you can't add more ";
return;
}
else {
cin >> car[carCount++];
Project: car rental system, Configuration: Debug x64 ------
oj -> C:\Users\Aya Atef\source\repos\car rental system\x64\Debug\car rental system.exe


roject Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
car rental system
Debug
x64
Local Windows Debugger
Auto
company.cpp
company.h
person.h
person.cpp
System.h
System.cpp*
customer.h
RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cpp
car.h
car rental system.cpp
car rental system
System
f 1 project)
55 else {
56 cin >> car[carCount++];
57 cout << "Do you want to add another car(y/n): ";
58 cin >> c;
59 }
60 }
61 while (c != 'n');
62 }
63 void System::addCustomer() {
64 int n;
65 c = 'n';
66 do {
67 if (customerCount == customerSize) {
68 cout << "\nYou exceeded the utmost number of customers. You can't add more." << endl;
69 return;
70 }
71 cout << "What kind of customer do you want to add? (1-person, 2-company): ";
72 cin >> n;
73 switch (n) {
74 case 1:
75 customer[customerCount] = new Person;
76 break;
77 case 2:
78 customer[customerCount] = new Company;
79 break;
80 default:
81 cout << "Invalid choice. Please try again." << endl;
82 continue;
83 }

company.cpp | company.h | person.h | person.cpp | System.h | System.cpp* | customer.h

RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | reservation.h | car.cpp

car rental system.cpp

car rental system | System

```
79              break;
80          default:
81              cout << "Invalid choice. Please try again." << endl;
82              continue;
83          }
84          cin >> *(customer[customerCount]);
85          customerCount++;
86          cout << "Do you want to add another customer? (y/n): ";
87          cin >> c;
88      } while (c != 'n');
89  }
90
91  void System::rentCar() {
92      c = 'n';
93      printline("##Rent car system");
94      do {
95          string s;
96          cout << "1-renting a car\n2-returning to main menu\n\tEnter choice:  ";
97          cin >> b;
98          if (b == 1) {
99              reservation.readCarId();
100             int carIndex = Search_car(reservation.getCarId());
101             if (carIndex == -1) {
102                 cout << "\nSorry!! Car specified isn't found";
103                 stat = notCompleted;
104                 return;
105             }
106             reservation.readCustomerName();
107             int customerIndex = search_customer(reservation.getCustomerId());
```

```
 rental system, Configuration: Debug x64 ------
ers\Aya Atef\source\repos\car rental system\x64\Debug\car rental system.exe
led, 0 up-to-date, 0 skipped ==========
```

Debug x64 Local Windows Debugger Auto

company.cpp | company.h | person.h | person.cpp | System.h | System.cpp* | customer.h | customer.cpp
RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | reservation.h | car.cpp | car.h | Date.h
car rental system.cpp

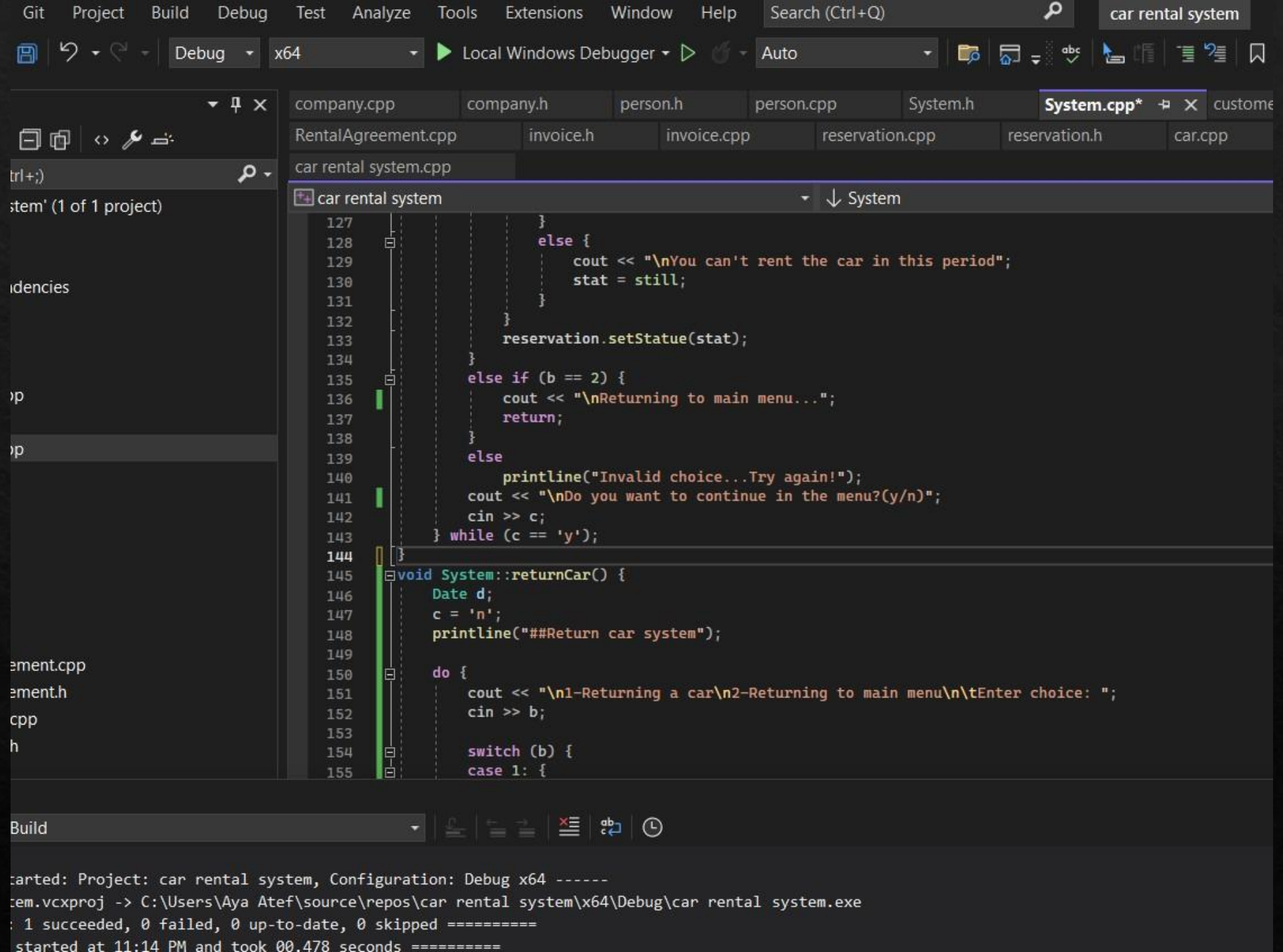
(Ctrl+;)
system' (1 of 1 project)
m
endencies
.cpp
.h
.cpp
.h
op
op
reement.cpp
reement.h
on.cpp
on.h
op

car rental system | System | rentCar()

```
103                 stat = notCompleted;
104                 return;
105             }
106             reservation.readCustomerName();
107             int customerIndex = search_customer(reservation.getCustomerId());
108             if (customerIndex == -1) {
109                 cout << "\nSorry!! Customer specified isn't found";
110                 stat = notCompleted;
111                 return;
112             }
113             else {
114                 cin >> reservation;
115                 if (car[carIndex].checkAvailability(reservation.getStartDate(), reservation.getEndDate())) {
116                     reservation.readInvoice();
117                     if (car[carIndex].checkSize() && customer[customerIndex]->checkSize()) {
118                         car[carIndex].addReservation(reservation);
119                         customer[customerIndex]->addReservation(reservation);
120                         cout << "\nReservation Completed" << endl;
121                         stat = completed;
122                     }
123                     else {
124                         cout << "\nfailed to rent";
125                         stat = still;
126                     }
127                 }
128                 else {
129                     cout << "\nYou can't rent the car in this period";
130                     stat = still;
131                 }
```

Build

started: Project: car rental system, Configuration: Debug x64 ------


Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
car rental system
Debug
x64
Local Windows Debugger
Auto
company.cpp
company.h
person.h
person.cpp
System.h
System.cpp*
RentalAgreement.cpp
invoice.h
invoice.cpp
reservation.cpp
reservation.h
car.cpp
car rental system.cpp
car rental system
System
127 }
128 else {
129 cout << "\nYou can't rent the car in this period";
130 stat = still;
131 }
132 }
133 reservation.setStatue(stat);
134 }
135 else if (b == 2) {
136 cout << "\nReturning to main menu...";
137 return;
138 }
139 else
140 printline("Invalid choice...Try again!");
141 cout << "\nDo you want to continue in the menu?(y/n)";
142 cin >> c;
143 } while (c == 'y');
144 }
145 void System::returnCar() {
146 Date d;
147 c = 'n';
148 printline("##Return car system");
149
150 do {
151 cout << "\n1-Returning a car\n2-Returning to main menu\n\tEnter choice: ";
152 cin >> b;
153
154 switch (b) {
155 case 1: {
Build
started: Project: car rental system, Configuration: Debug x64 ------
tem.vcxproj -> C:\Users\Aya Atef\source\repos\car rental system\x64\Debug\car rental system.exe
1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
started at 11:14 PM and took 00.478 seconds ==========

1 project)

RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | reservation.h | car.cpp

car rental system.cpp

car rental system | System

```
139             else
140                 printline("Invalid choice...Try again!*");
141             cout << "\nDo you want to continue in the menu?(y/n)";
142             cin >> c;
143         } while (c == 'y');
144     }
145     void System::returnCar() {
146         Date d;
147         c = 'n';
148         printline("##Return car system");
149
150         do {
151             cout << "\n1-Returning a car\n2-Returning to main menu\n\tEnter choice: ";
152             cin >> b;
153
154             switch (b) {
155             case 1: {
156                 int id;
157                 cout << "\nEnter the ID of the car you want to return: ";
158                 cin >> id;
159
160                 int index = Search_car(id);
161                 if (index == -1) {
162                     cout << "\nSorry! The specified car is not found." << endl;
163                     stat = still;
164                 }
165                 else {
166                     for (int i = index; i < carCount - 1; i++) {
167                         car[i] = car[i + 1];
168                     }
169                     carCount--;
170
171                     stat = canceled;
172                     cout << "\nCar returned successfully." << endl;
173                 }
174
175                 reservation.setStatue(stat);
176                 break;
177             }
178
179             case 2:
180                 cout << "\nReturning to the main menu..." << endl;
181                 return;
182
183             default:
184                 cout << "\nInvalid choice. Please try again!" << endl;
185             }
186
187             cout << "\nDo you want to continue in the menu? (y/n): ";
188             cin >> c;
189         } while (c == 'y');
190     }
191
192     void System::viewAllAvailableCars() {
193         check = false;
194         c = 'n';
195         printline("##Viewing available cars");
196         do {
197            std:: cout << "\n1-View available cars\n2-Return to main menu\n\tEnter choice: ";
198             std::cin >> b;
199             if (b == 1) {
200                 Date start_date, end_date;
201                 cout << "Enter rental start date: " << endl;
202                 cin >> start_date;
```

company.cpp | company.h | person.h | person.cpp | System.h | System.cpp*

RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp | reservation.h | car.c

car rental system.cpp

car rental system | System

```
190 }
191
192 void System::viewAllAvailableCars() {
193     check = false;
194     c = 'n';
195     printline("##Viewing available cars");
196     do {
197         std:: cout << "\n1-View available cars\n2-Return to main menu\n\tEnter choice: ";
198         std::cin >> b;
199         if (b == 1) {
200             Date start_date, end_date;
201             cout << "Enter rental start date: " << endl;
202             cin >> start_date;
203             cout << "Enter rental end date: " << endl;
204             cin >> end_date;
205             for (int index = 0; index < carCount; index++)
206             {
207                 if (car[index].checkAvailability(start_date, end_date)) {
208                     cout << "\nCar details:\n" << car[index];
209                     check = true;
210                     break;
211                 }
212             }if (check == false)
213                 cout << "\nNo available cars found...";
214         }
215         else if (b == 2) {
216             cout << "\nReturning to main menu...";
217             return;
218         }
219         else cout << "\nInvalid choice...Try again!";
220         cout << "\nContinue in the menu?(y/n)";
221         cin >> c;
222     } while (c == 'y');
223 }
224     void System::viewCustomerRentalHistory(){
225         printline("##Viewing Rental history");
226         cout << "\n1-View Rental history\n2-Return to main menu\n\tEnter choice: ";
227         cin >> b;
228         if (b == 1) {
229             int id;
230             cout << "Enter Customers id: " << endl;
231             cin >> id;
232             cout << endl;
233             int index = search_customer(id);
234             if (index == -1)
235                 cout << "Customer not found" << endl;
236             else
237                 customer[index]->printData(cout) << endl;
238         }
239         else if (b == 2) {
240             cout << "\nReturning to main menu...";
241             return;
242         }
243         else cout << "\nInvalid choice...Try again!";
244     }
245     void System::generateReports() {
246         printline("##Generate reports");
247         cout << "\n1-Report car\n2-Customer history\n3-Return to main menu\n\tEnter choice: ";
248         cin >> b;
249         switch (b) {
250         case 1:
251             if (customerCount != 0) {
252                 cout << setw(50) << "Customers rental history:" << endl;
253                 for (int i = 0; i < customerCount; i++)
```

company.cpp | company.h | person.h | person.cpp | System.h

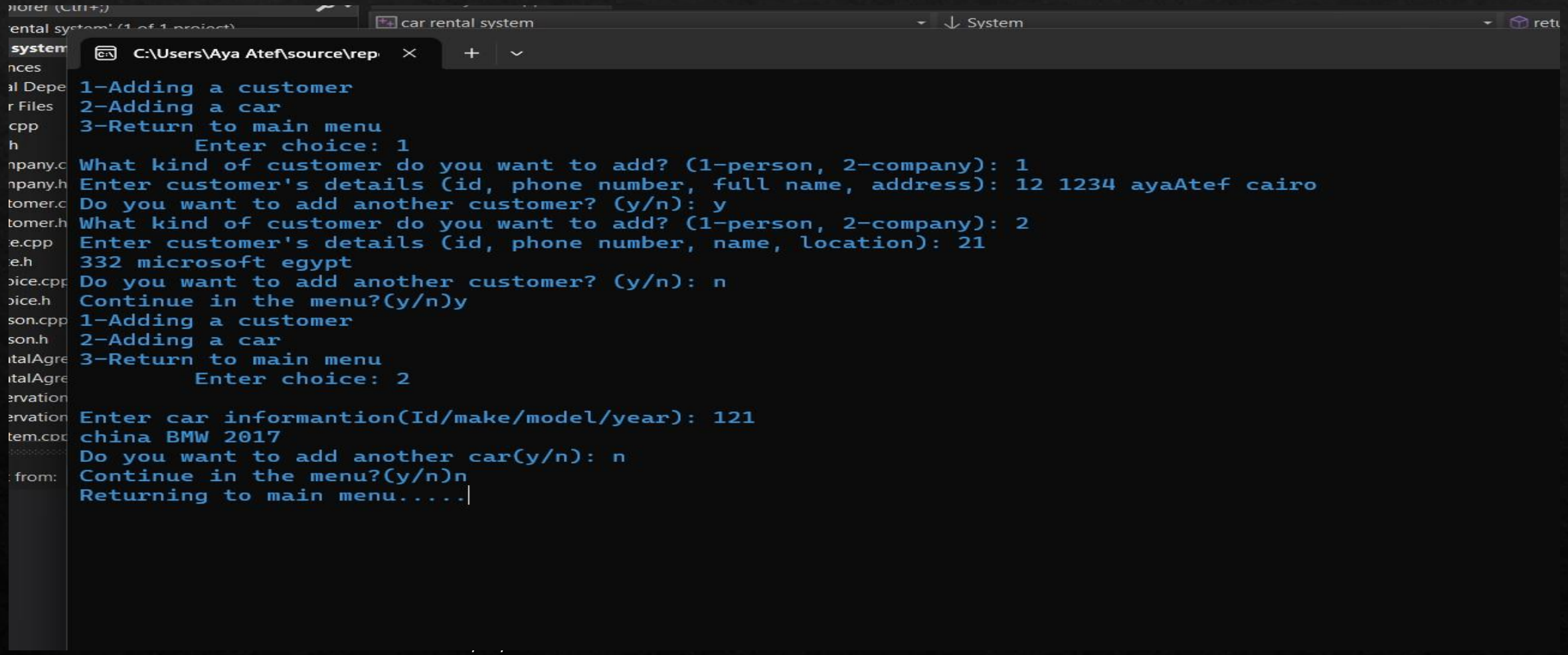
RentalAgreement.cpp | invoice.h | invoice.cpp | reservation.cpp

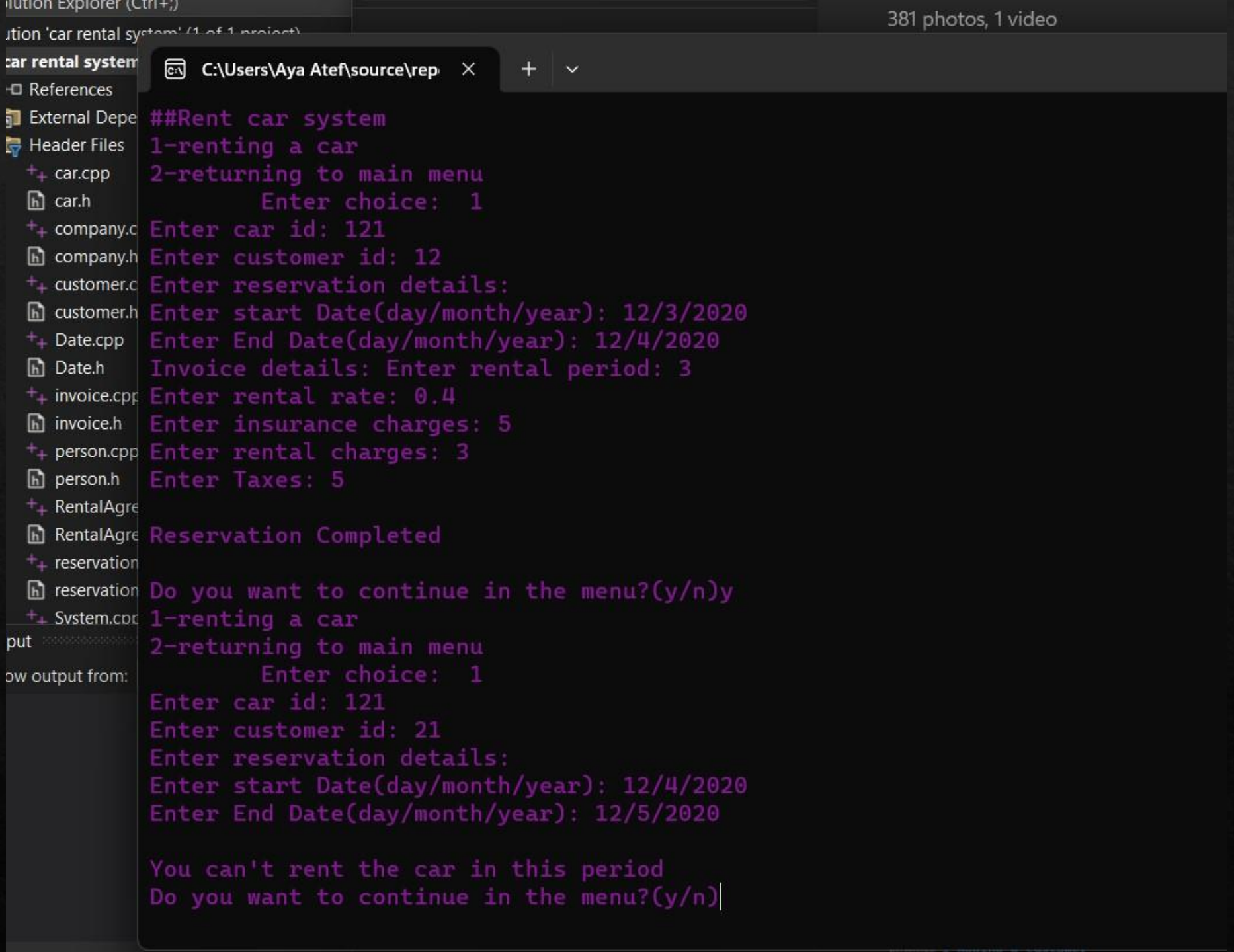
car rental system.cpp

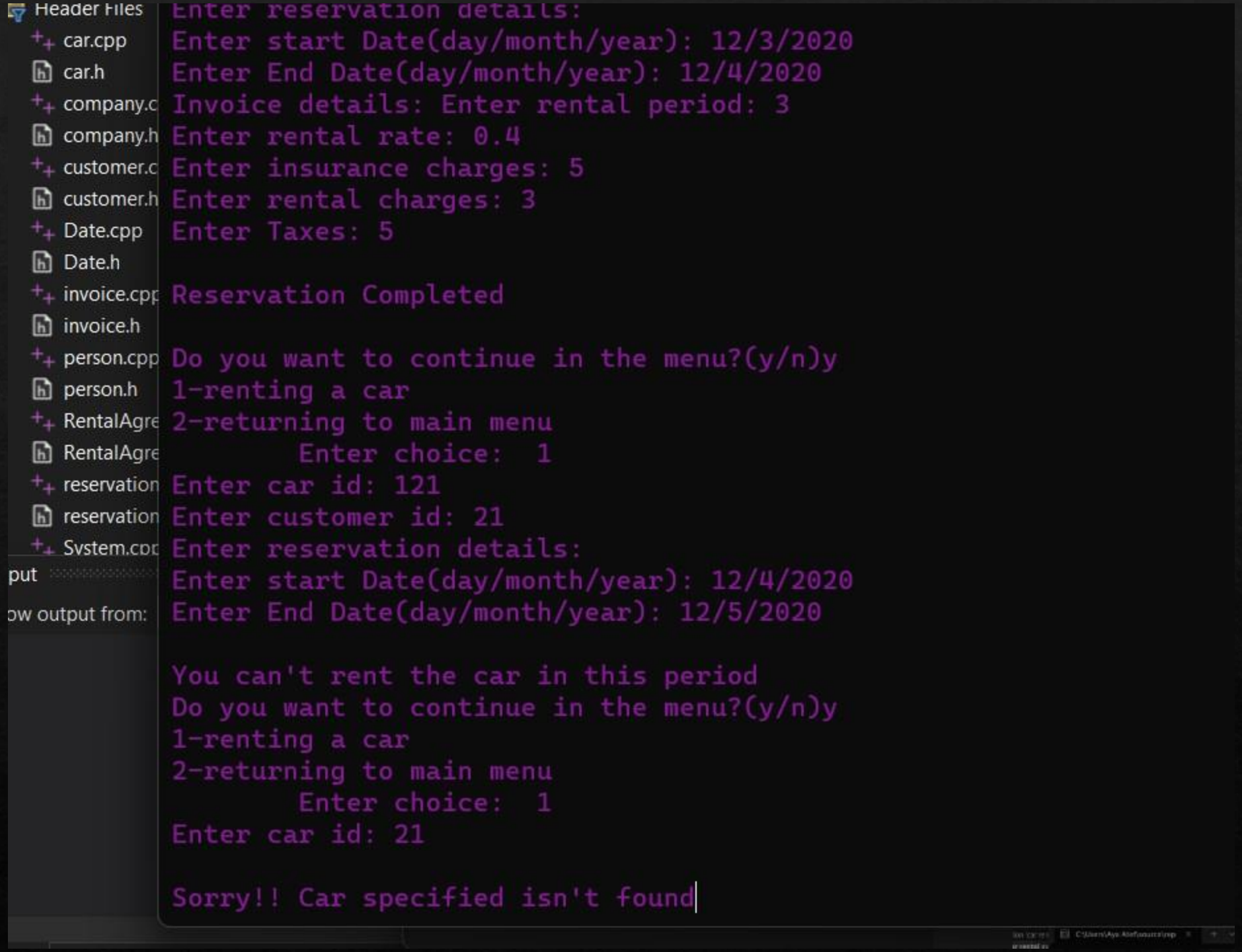
car rental system ▾ | System

```
238         }
239         else if (b == 2) {
240             cout << "\nReturning to main menu...";
241             return;
242         }
243         else cout << "\nInvalid choice...Try again!";
244     }
245     void System::generateReports() {
246         printline("##Generate reports");
247         cout << "\n1-Report car\n2-Customer history\n3-Return to main menu\n\tEnter choice: ";
248         cin >> b;
249         switch (b) {
250         case 1:
251             if (customerCount != 0) {
252                 cout << setw(50) << "Customers rental history:" << endl;
253                 for (int i = 0; i < customerCount; i++)
254                 {
255                     cout << "---Customer number" << i + 1 << " details: " << endl;
256                     customer[i]->printData(cout) << endl;
257                 }
258             }
259             break;
260         case 2:
261             if (carCount != 0) {
262                 cout << setw(50) << "Cars rental history: " << endl;
263                 for (int i = 0; i < carCount; i++)
264                 {
265                     cout << "---Car number " << i + 1 << " details: " << endl;
266                     cout << car[i] << endl;
267                 }
268             }
269             break;
270         case 3:
271             cout << "\nReturning to main menu...";
272             return;
273         default:
274             cout << "\nInvalid choice...Try again ";
275         }
276     }
277
278     System::~System(){
279         delete[]car;
280         for (int i = 0; i < customerCount; i++)
281             delete customer[i];
282         delete[]customer;
283     }
```

Screen shots for the output

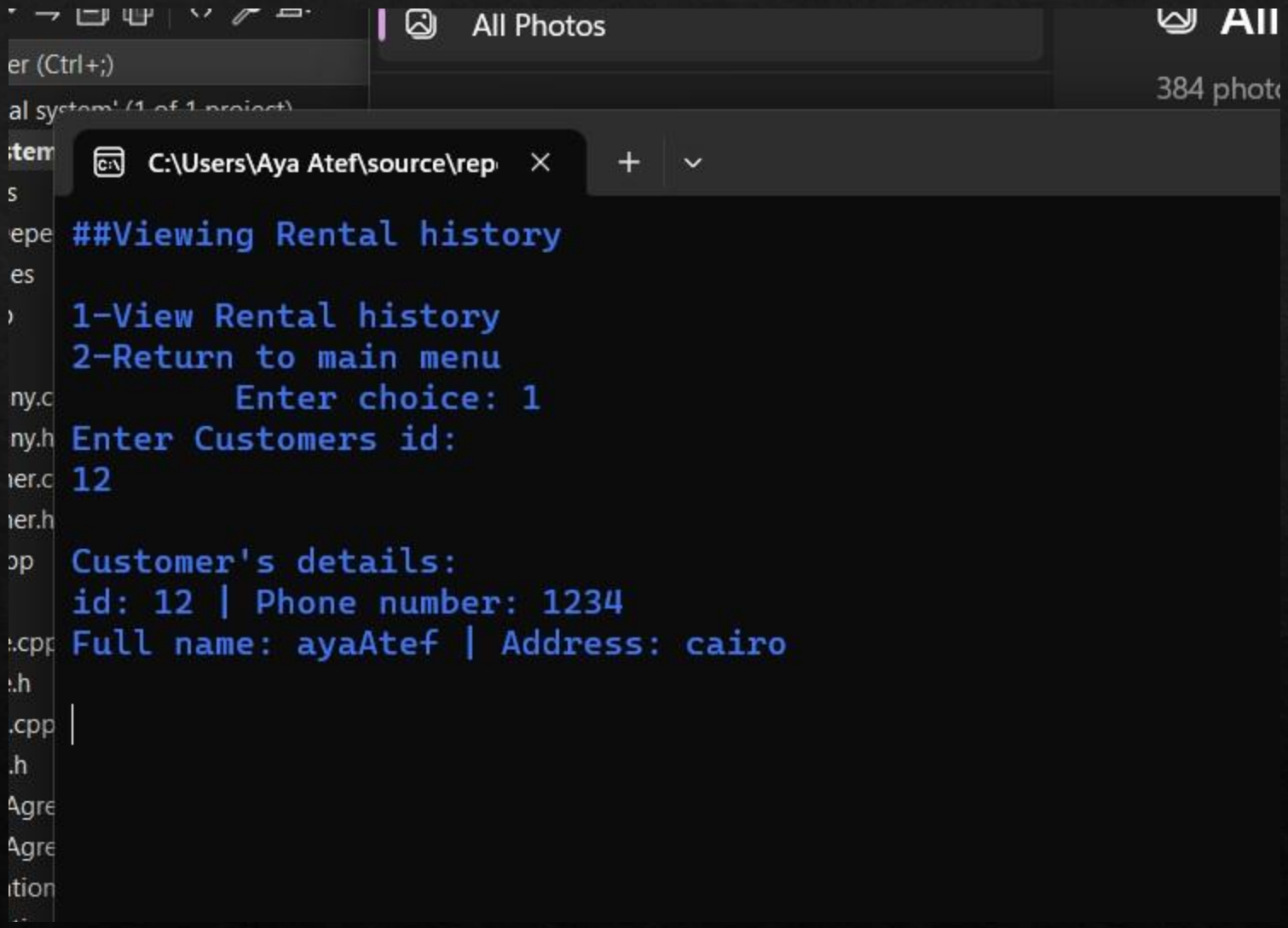


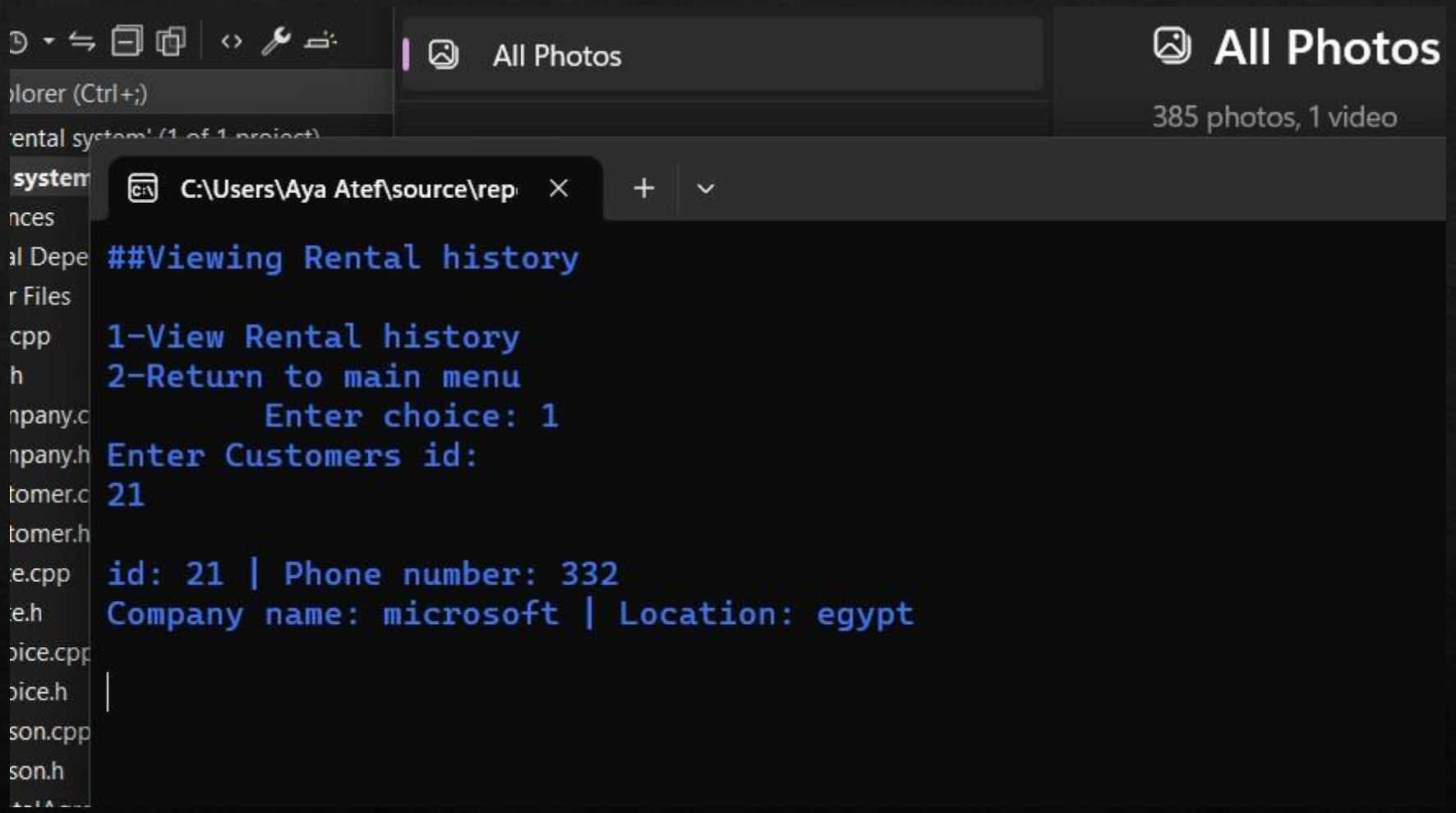

381 photos, 1 video
References
External Depe
Header Files
car.cpp
car.h
company.c
company.h
customer.c
customer.h
Date.cpp
Date.h
invoice.cpp
invoice.h
person.cpp
person.h
RentalAgre
RentalAgre
reservation
reservation
System.cpp
C:\Users\Aya Atef\source\rep
##Rent car system
1-renting a car
2-returning to main menu
Enter choice: 1
Enter car id: 121
Enter customer id: 12
Enter reservation details:
Enter start Date(day/month/year): 12/3/2020
Enter End Date(day/month/year): 12/4/2020
Invoice details: Enter rental period: 3
Enter rental rate: 0.4
Enter insurance charges: 5
Enter rental charges: 3
Enter Taxes: 5
Reservation Completed
Do you want to continue in the menu?(y/n)y
1-renting a car
2-returning to main menu
Enter choice: 1
Enter car id: 121
Enter customer id: 21
Enter reservation details:
Enter start Date(day/month/year): 12/4/2020
Enter End Date(day/month/year): 12/5/2020
You can't rent the car in this period
Do you want to continue in the menu?(y/n)

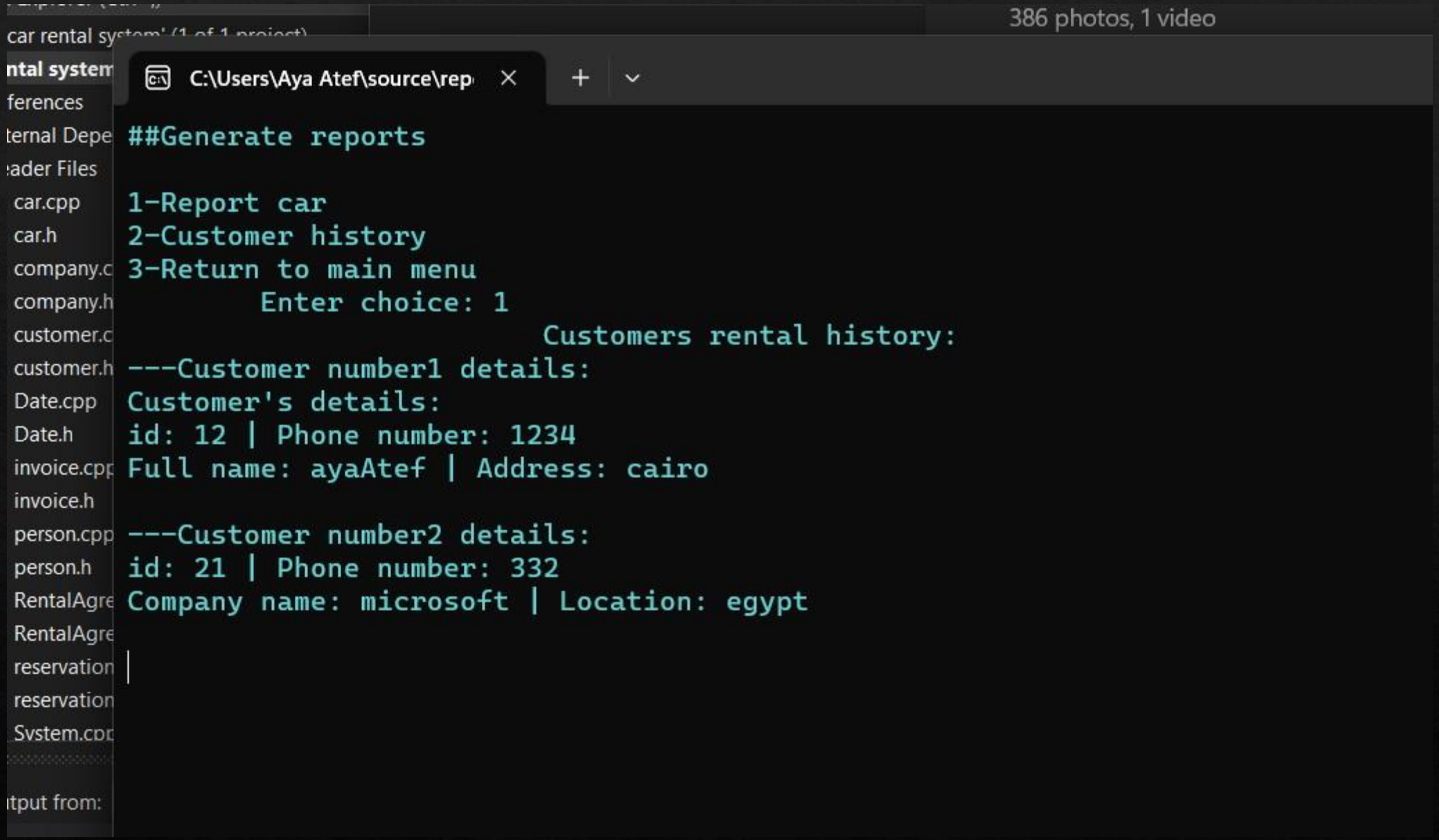

Header Files
car.cpp
car.h
company.c
company.h
customer.c
customer.h
Date.cpp
Date.h
invoice.cpp
invoice.h
person.cpp
person.h
RentalAgre
RentalAgre
reservation
reservation
System.cpp
put
ow output from:
Enter reservation details:
Enter start Date(day/month/year): 12/3/2020
Enter End Date(day/month/year): 12/4/2020
Invoice details: Enter rental period: 3
Enter rental rate: 0.4
Enter insurance charges: 5
Enter rental charges: 3
Enter Taxes: 5
Reservation Completed
Do you want to continue in the menu?(y/n)y
1-renting a car
2-returning to main menu
Enter choice: 1
Enter car id: 121
Enter customer id: 21
Enter reservation details:
Enter start Date(day/month/year): 12/4/2020
Enter End Date(day/month/year): 12/5/2020
You can't rent the car in this period
Do you want to continue in the menu?(y/n)y
1-renting a car
2-returning to main menu
Enter choice: 1
Enter car id: 21
Sorry!! Car specified isn't found

C:\Users\Aya Atef\source\rep

```
##Viewing available cars

1-View available cars
2-Return to main menu
        Enter choice: 1
Enter rental start date:
12/6/2020
Enter rental end date:
12/7/2020

Car details:

        =====CAR INFORMATION=====
Id: 121 | Make: china | model : BMW | Year : 2017     ***Reservation 1 ***

Start Date: 12/3/2020  |End Date: 12/4/2020
Invoice details:
 Rental period: 3 |Rental Price: 6.2 |Insurance charges: 5 |Rental rate: 0.4 |rentalCharges: 3 | Taxes: 5
Customer id: 12
Continue in the menu?(y/n)
```


All Photos
384 photo
C:\Users\Aya Atef\source\rep
##Viewing Rental history
1-View Rental history
2-Return to main menu
Enter choice: 1
Enter Customers id:
12
Customer's details:
id: 12 | Phone number: 1234
Full name: ayaAtef | Address: cairo

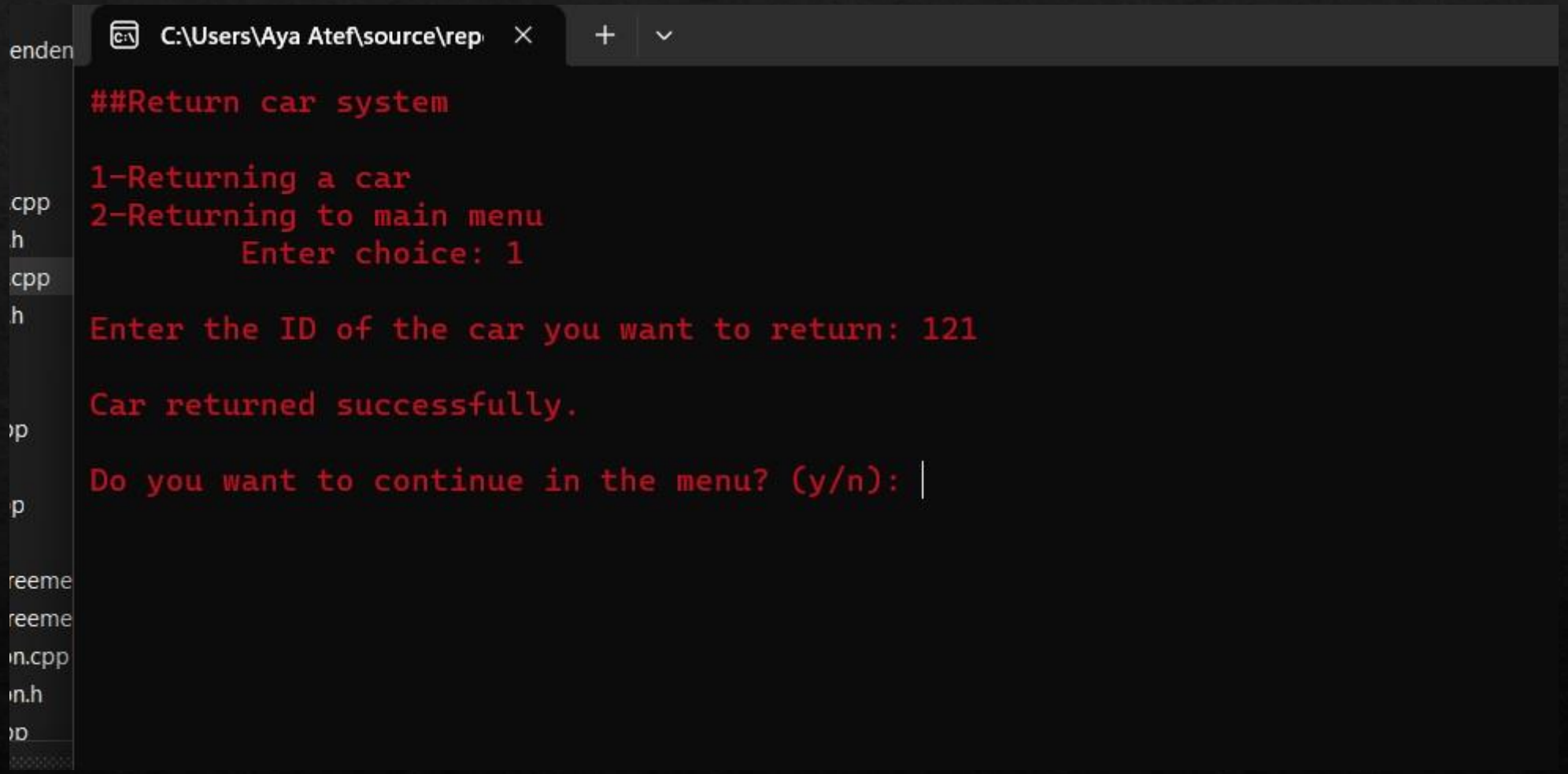

All Photos
All Photos
385 photos, 1 video
C:\Users\Aya Atef\source\rep
##Viewing Rental history
1-View Rental history
2-Return to main menu
Enter choice: 1
Enter Customers id:
21
id: 21 | Phone number: 332
Company name: microsoft | Location: egypt


386 photos, 1 video
C:\Users\Aya Atef\source\rep
ntal system
ferences
ternal Depe
ader Files
car.cpp
car.h
company.c
company.h
customer.c
customer.h
Date.cpp
Date.h
invoice.cpp
invoice.h
person.cpp
person.h
RentalAgre
RentalAgre
reservation
reservation
System.cpp
tput from:
##Generate reports
1-Report car
2-Customer history
3-Return to main menu
Enter choice: 1
Customers rental history:
---Customer number1 details:
Customer's details:
id: 12 | Phone number: 1234
Full name: ayaAtef | Address: cairo
---Customer number2 details:
id: 21 | Phone number: 332
Company name: microsoft | Location: egypt

C:\Users\Aya Atef\source\rep

```
##Generate reports

1-Report car
2-Customer history
3-Return to main menu
        Enter choice: 2
                            Cars rental history:
---Car number 1 details:

        =====CAR INFORMATION=====
Id: 121 | Make: china | model : BMW | Year : 2017     ***Reservation 1 ***

Start Date: 12/3/2020  |End Date: 12/4/2020
Invoice details:
 Rental period: 3 |Rental Price: 6.2 |Insurance charges: 5 |Rental rate: 0.4 |rentalCharges: 3 | Taxes: 5
Customer id: 12
```

```
1-Adding a customer
2-Adding a car
3-Return to main menu
        Enter choice: 1
What kind of customer do you want to add? (1-person, 2-company): 1
Enter customer's details (id, phone number, full name, address): 12 123 ayaAtef egypt
Do you want to add another customer? (y/n): n
Continue in the menu?(y/n)y
1-Adding a customer
2-Adding a car
3-Return to main menu
        Enter choice: 2

Enter car informantion(Id/make/model/year): 121 china BMW 2020
Do you want to add another car(y/n): n
Continue in the menu?(y/n)n
Returning to main menu.....
```


C:\Users\Aya Atef\source\rep
##Return car system
1-Returning a car
2-Returning to main menu
Enter choice: 1
Enter the ID of the car you want to return: 121
Car returned successfully.
Do you want to continue in the menu? (y/n):

Summary of what happens in the Program:

when I open the program it takes me to the main menu of the program which shows a list of the other menus of the program and each menu handle different things in my system .

first it handles the menu of adding either cars or customers:

This menu asks the user if he want to add a car or a customer if he want to add a customer he press 1 either if he want to add a car he press 2 then if he presses 1 the program asks him to enter the details of the customer like the name , address and phone number these details will be stored in my system then it asks him if he want to add another customer if not he asks him whether to still in this menu or return back to the main menu if he said he want to still in this menu then the program asks him again if he want to add a customer or a car

let's say he chose a car then the system asks him to add the details of the car like :id,model,make and the year then stores them in memory and repeats asking to add more cars or stil in the menu if he want to add customers now i will exit this menu yo return to move me to the second meny this menu all its functionality is to rent a car 2 back to the main menu of the program i press to enter the id of the car if found it asks to enter the name of the customer and if not it gives me from the system it asks the user error message that the car isn't found then if found i enter the name of the customer and checks if it is found in my array or not if not it gives me error message that the customer isn't found and then exist the menu but if found it asks to enter the start and end date of my reservation then checks if there isoverlapping if not it allows me to enter the money for the reservation

Now I exit this menu and return to the main menu
I press 3 to move me to the third menu where i can return the car after the period finishes and then it gives m message that the car has been returned successfully
I exit this menu and returns to the main menu then I press 4 it moves me to the fourth where I can view all available cars in a specific period .it asks me to enter the start date and the end date then it prints all available cars in this period
I exit this menu to return to the main menu I press 5 it moves me to the view rental history menu which shows the for every customer his reservations
I exit this menu and return to the main menu
then I press 6 this menu prints all customers and cars in the system and their reservations

Thank you I hope you nice day and goos time:)