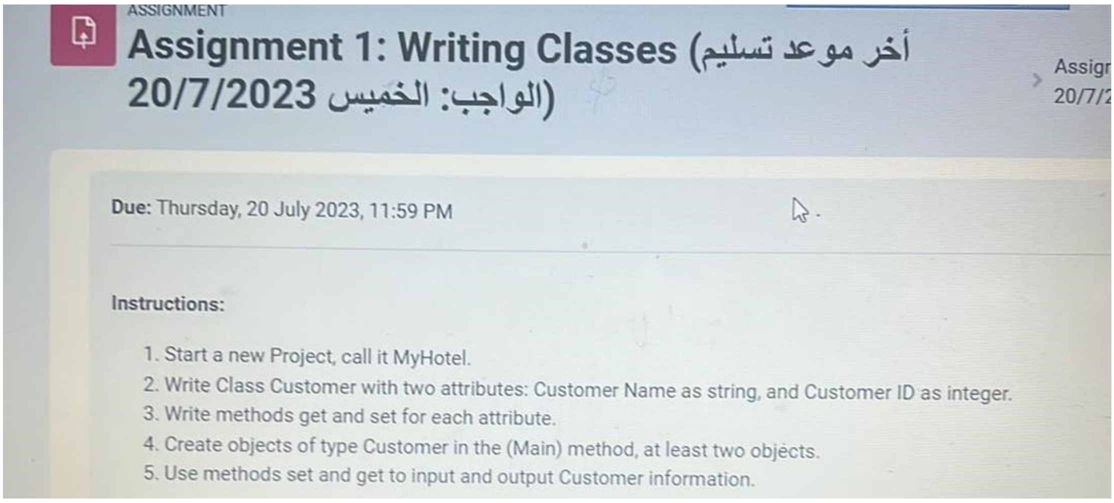
Assignment 1 Hotel



ASSIGNMENT

Assignment 1: Writing Classes (أخر موعد تسليم
الواجب: الخميس 20/7/2023)

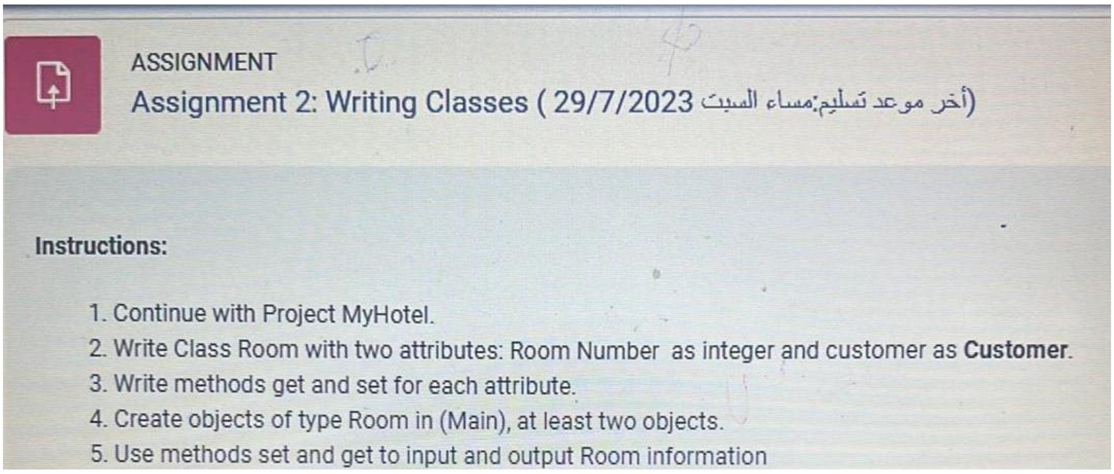
Due: Thursday, 20 July 2023, 11:59 PM

Instructions:

1. Start a new Project, call it MyHotel.
2. Write Class Customer with two attributes: Customer Name as string, and Customer ID as integer.
3. Write methods get and set for each attribute.
4. Create objects of type Customer in the (Main) method, at least two objects.
5. Use methods set and get to input and output Customer information.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment1
{
    class Program
    {
        static void Main(string[] args)
        {
            Customer customer1 = new Customer();
            Customer customer2 = new Customer();

            customer1.SetCustomerName("Mohammed");
            customer1.SetCustomerID(1001);

            customer2.SetCustomerName("Mahmoud");
            customer2.SetCustomerID(1002);
            Console.WriteLine($"Customer 1 Name: {customer1.GetCustomerName()}, ID: {customer1.GetCustomerID()}");
            Console.WriteLine($"Customer 2 Name: {customer2.GetCustomerName()}, ID: {customer2.GetCustomerID()}");
        }
    }
    class Customer
    {
        private string customerName;
        private int customerID;

        public string GetCustomerName()
        {
            return customerName;
        }

        public void SetCustomerName(string name)
        {
            customerName = name;
        }

        public int GetCustomerID()
        {
            return customerID;
        }

        public void SetCustomerID(int id)
        {
            customerID = id;
        }
    }
}
```

Assignment 2 Hotel



ASSIGNMENT

Assignment 2: Writing Classes (29/7/2023 أخر موعد تسليم:مساء السبت)

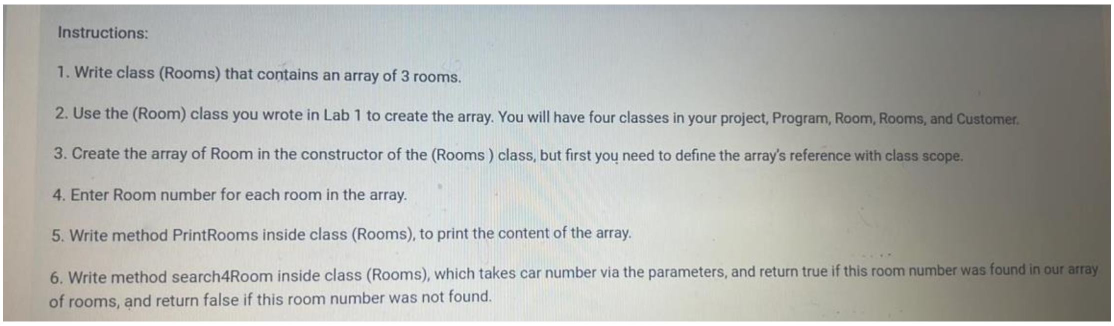
Instructions:

1. Continue with Project MyHotel.
2. Write Class Room with two attributes: Room Number as integer and customer as **Customer**.
3. Write methods get and set for each attribute.
4. Create objects of type Room in (Main), at least two objects.
5. Use methods set and get to input and output Room information

```
using System;


namespace MyHotel
{
    class Program
    {
        static void Main(string[] args)
        {
            Customer customer1 = new Customer();
            Customer customer2 = new Customer();


            customer1.SetCustomerName("Ahmed");
            customer1.SetCustomerID(749625);


            customer2.SetCustomerName("Yaser");
            customer2.SetCustomerID(529669);


            Console.WriteLine("Customer1 Name: {0}, Customer1 ID: {1}", customer1.GetCustomerName(), customer1.GetCustomerID());
            Console.WriteLine("Customer2 Name: {0}, Customer2 ID: {1}", customer2.GetCustomerName(), customer2.GetCustomerID());
    Room room1 = new Room();
    Room room2 = new Room();

    Console.WriteLine("RoomNumber: {0}, Customer1 : {1}", 101, "Ahmed Rami");
    Console.WriteLine("RoomNumber: {0}, Customer2 : {1}", 202, "Yaser Mahmoud");

  }
}

class Customer
{
  private string customerName;
  private int customerID;

  public string GetCustomerName()
  {
    return customerName;
  }

  public void SetCustomerName(string name)
  {
    customerName = name;
  }

  public int GetCustomerID()
  {
    return customerID;
  }

  public void SetCustomerID(int id)
  {
```

```
            customerID = id;
        }
    }
    class Room
    {
        private int RoomNumber;
        private Customer customer;

        public int GetRoomNumber()
        {
            return RoomNumber;
        }
        public void SetRoomnumber(int room)
        {
            RoomNumber = room;
        }
        public Customer Getcustomer1()
        {
            return customer;
        }
        public void Setcustomer(Customer customer1)
        {
            customer = customer1;
        }
    }
}
```

Assignment 3 Hotel



Instructions:

1. Write class (Rooms) that contains an array of 3 rooms.

2. Use the (Room) class you wrote in Lab 1 to create the array. You will have four classes in your project, Program, Room, Rooms, and Customer.

3. Create the array of Room in the constructor of the (Rooms) class, but first you need to define the array's reference with class scope.

4. Enter Room number for each room in the array.

5. Write method PrintRooms inside class (Rooms), to print the content of the array.

6. Write method search4Room inside class (Rooms), which takes car number via the parameters, and return true if this room number was found in our array of rooms, and return false if this room number was not found.

```
using System;


namespace MyHotel
{
  class Program
  {
    static void Main(string[] args)
    {
      Rooms rooms = new Rooms();
      while (true)
      {
        Console.WriteLine("Main Menu ");
        Console.WriteLine("1.Print Rooms");
        Console.WriteLine("2.Search Rooms");
        Console.WriteLine("3.Exit");
        Console.WriteLine(" Enter your choice :");
        string input = Console.ReadLine();
        switch (input)
        {
          case "1":
            rooms.PrintRooms();
            break;
          case "2":
            Console.WriteLine("Enter a room number to search for:");
```

```
                    int roomNumberToSearch = int.Parse(Console.ReadLine());
                    bool roomFound = rooms.Search4Room(roomNumberToSearch);
                    if (roomFound)
                    {
                        Console.WriteLine("Room is Available");
                    }
                    else
                    {
                        Console.WriteLine("Room is not Available");
                    }
                    break;
                case "0":
                    return;
                default:
                    Console.WriteLine("Invalid input. Please try again.");
                    break;
            }
            Console.WriteLine();
        }
    }
}
class Customer
{
    private string customerName;
    private int customerID;
    public void SetCustomerName(string name)
    {
        customerName = name;
    }
    public string GetCustomerName()
    {
        return customerName;
    }
    public void SetCustomerID(int id)
    {
        customerID = id;
    }
    public int GetCustomerID()
    {
        return customerID;
    }
}
class Room
{
    private int RoomNumber;
    private Customer customer;
    public int GetRoomNumber()
    {
        return RoomNumber;
    }
    public void SetRoomnumber(int room)
    {
        RoomNumber = room;
    }
    public Customer Getcustomer1()
    {
        return customer;
    }
    public void Setcustomer(Customer customer1)
    {
        customer = customer1;
    }
```

```
}
class Rooms
{
    private Room[] rooms;
    public Rooms()
    {
        rooms = new Room[3];
        for (int i = 0; i < 3; i++)
        {
            rooms[i] = new Room();
        }
        rooms[0].SetRoomnumber(5);
        rooms[1].SetRoomnumber(6);
        rooms[2].SetRoomnumber(7);
    }
    public void PrintRooms()
    {
        for (int i = 0; i < 3; i++)
        {
            Console.WriteLine("Room Number: {0}", rooms[i].GetRoomNumber());
            if (rooms[i].Getcustomer1() != null)
            {
                string customername = rooms[i].Getcustomer1().GetCustomerName();
                int customerid = rooms[i].Getcustomer1().GetCustomerID();
                Console.WriteLine("customer name: {0}", customername);
                Console.WriteLine("customer id: {0}", customerid);
            }
        }
    }

    public bool Search4Room(int roomNumber)
```

```
        {
            for (int i = 0; i < 3; i++)
            {
                if (rooms[i].GetRoomNumber() == roomNumber)
                {
                    if (rooms[i].Getcustomer1() != null)
                    {
                        string customername = rooms[i].Getcustomer1().GetCustomerName();
                        int customerid = rooms[i].Getcustomer1().GetCustomerID();
                        Console.WriteLine("customer name: {0}", customername);
                        Console.WriteLine("customer id: {0}", customerid);
                    }
                    return true;
                }
            }
            return false;
        }
    }
}
```

Assignment 5 Hotel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace myhotell
{
    class Program
    {
        static void Main(string[] args)
        {
            Customer customer1 = new Customer();
            Customer customer2 = new Customer();
            Rooms room = new Rooms();

            customer1.SetCustomerName("khaled");
            customer1.SetCustomerID(15448);

            customer2.SetCustomerName("mohammad");
            customer2.SetCustomerID(54878);



            Console.WriteLine("Customer1 Name: {0}, Customer1 ID: {1}", customer1.GetCustomerName(), customer1.GetCustomerID());
            Console.WriteLine("Customer2 Name: {0}, Customer2 ID: {1}", customer2.GetCustomerName(), customer2.GetCustomerID());



            Room room1 = new Room();
```

```
   Room room2 = new Room();

   Console.WriteLine("RoomNumber: {0}, Custoername : {1}", 505, "khaled yousef");
   Console.WriteLine("RoomNumber: {0}, Customername : {1}", 909, "mohammad
hashem ");

   while (true)
   {
     Console.WriteLine("My Hotel\n ");
     Console.WriteLine("Main Menu\n ");
     Console.WriteLine("1. Reserve Room\r\n");
     Console.WriteLine("2. Return Room");
     Console.WriteLine("3. Search 4 Room");
     Console.WriteLine("4. Print List of Rooms");
     Console.WriteLine("5. Exit");
     Console.WriteLine(" Enter your choice :");
     string input = Console.ReadLine();

     switch (input)
     {
       case "1":
         room.RentRoom();
         break;
       case "2":
         room.ReturnRoom();
         break;
       case "3":
         Console.WriteLine("Enter a room number to search for:");
         int roomNumberToSearch = int.Parse(Console.ReadLine());
         bool roomFound = room.Search4Room(roomNumberToSearch);
         if (roomFound)
```

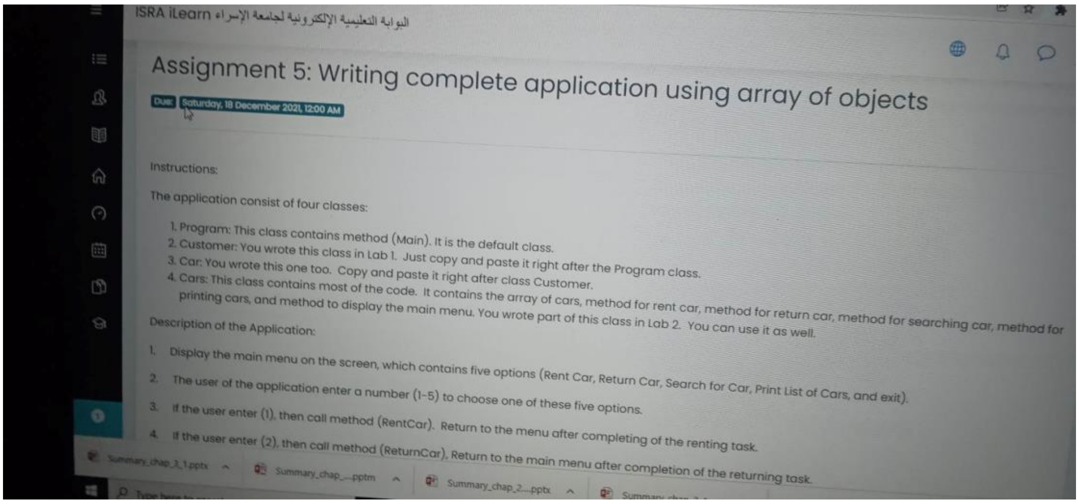
```
                    {
                        Console.WriteLine("Room is Available");
                    }
                    else
                    {
                        Console.WriteLine("Room is not Available");
                    }
                    break;
                case "4":
                    room.PrintRooms();
                    break;
                case "5":
                    return;
                default:
                    Console.WriteLine("Invalid input. Please try again.");
                    break;
            }
            Console.WriteLine();
        }
    }
}

class Customer
{
    private string customerName;
    private int customerID;
    public string GetCustomerName()
    {
        return customerName;
    }
    public void SetCustomerName(string name)
    {
        customerName = name;
    }

    public int GetCustomerID()
    {
        return customerID;
    }

    public void SetCustomerID(int id)
    {
        customerID = id;
    }
}
class Room
{
    private int RoomNumber;
    private Customer customer;

    public int GetRoomNumber()
    {
        return RoomNumber;
    }
    public void SetRoomnumber(int room)
    {
        RoomNumber = room;
    }
    public Customer Getcustomer1()
    {
        return customer;
    }
    public void Setcustomer(Customer customer1)
    {
        customer = customer1;
    }
}


class Rooms
{
    private Room[] rooms;
    public Rooms()
    {
        rooms = new Room[3];
        for (int i = 0; i < 3; i++)
        {
            rooms[i] = new Room();
        }
        rooms[0].SetRoomnumber(5);
        rooms[1].SetRoomnumber(6);
        rooms[2].SetRoomnumber(7);
    }
    public void RentRoom()
    {
        Console.WriteLine("My Hotel");
        Console.WriteLine("Rent Room");
        Console.Write("Enter Customer ID: ");
        int customerID = int.Parse(Console.ReadLine());
        Console.Write("Enter Room No: ");
        int roomNumber = int.Parse(Console.ReadLine());
```

```
Room roomToRent = null;
for (int i = 0; i < rooms.Length; i++)
{
    if (rooms[i].GetRoomNumber() == roomNumber)
    {
        roomToRent = rooms[i];
        break;
    }
}

if (roomToRent != null)
{
    Customer customer = roomToRent.Getcustomer1();
    if (customer != null)
    {
        Console.WriteLine("Customer Name: {0}", customer.GetCustomerName());
    }
    else
    {
        Console.WriteLine("No customer assigned to this room.");
    }

    Console.WriteLine("Room Number: {0}", roomToRent.GetRoomNumber());
    Console.Write("Correct? (Y/N) ");
    string confirmation = Console.ReadLine();
    if (confirmation.Equals("Y", StringComparison.OrdinalIgnoreCase))
    {
        // Perform the renting task (you need to implement this logic)
        if (customer == null)
        {
            Customer newCustomer = new Customer();
```

```
                    newCustomer.SetCustomerID(customerID);
                    // Here, you might want to fetch the customer's name from some source
                    newCustomer.SetCustomerName("Sample Customer");
                    roomToRent.Setcustomer(newCustomer); // Assign the customer to the room
                    Console.WriteLine("Room rented successfully!");
                }
                else
                {
                    Console.WriteLine("Room is already rented by a customer.");
                }
            }
        }
        else
        {
            Console.WriteLine("Room not found.");
        }
    }
    public void ReturnRoom()
    {
        Console.WriteLine("My Hotel");
        Console.WriteLine("Return Room");
        Console.Write("Enter Room No: ");
        int roomNumber = int.Parse(Console.ReadLine());

        Room roomToReturn = null;
        for (int i = 0; i < rooms.Length; i++)
        {
            if (rooms[i].GetRoomNumber() == roomNumber)
            {
                roomToReturn = rooms[i];
                break;
            }
        }

        if (roomToReturn != null)
        {
            Customer customer = roomToReturn.Getcustomer1();
            if (customer != null)
            {
                Console.WriteLine("Customer Name: {0}", customer.GetCustomerName());
            }
            else
            {
                Console.WriteLine("No customer assigned to this room.");
            }

            Console.WriteLine("Room Number: {0}", roomToReturn.GetRoomNumber());
            Console.Write("Correct? (Y/N) ");
            string confirmation = Console.ReadLine();
            if (confirmation.Equals("Y", StringComparison.OrdinalIgnoreCase))
            {
                // Perform the returning task (you need to implement this logic)
                if (customer != null)
                {
                    roomToReturn.Setcustomer(null);  // Remove the customer from the room
                    Console.WriteLine("Room returned successfully!");
                }
                else
                {
                    Console.WriteLine("No customer assigned to this room. Cannot return.");
                }
            }
        }
        else
        {
            Console.WriteLine("Room not found.");
        }
    }

    public void PrintRooms()
    {
        for (int i = 0; i < 3; i++)
        {
            Console.WriteLine("Room Number: {0}", rooms[i].GetRoomNumber());
            if (rooms[i].Getcustomer1() != null)
            {
                string customername = rooms[i].Getcustomer1().GetCustomerName();
                int customerid = rooms[i].Getcustomer1().GetCustomerID();
                Console.WriteLine("Customer Name: {0}", customername);
                Console.WriteLine("Customer ID: {0}", customerid);
            }
        }
    }

    public bool Search4Room(int roomNumber)
    {
        for (int i = 0; i < 3; i++)
        {
            if (rooms[i].GetRoomNumber() == roomNumber)
            {
                if (rooms[i].Getcustomer1() != null)
                {
```

```
                        string customername = rooms[i].Getcustomer1().GetCustomerName();
                        int customerid = rooms[i].Getcustomer1().GetCustomerID();
                        Console.WriteLine("Customer Name: {0}", customername);
                        Console.WriteLine("Customer ID: {0}", customerid);
                    }
                    return true;
                }
            }
            return false;
        }
    }
}
```

Assignment 5 car



Assignment 5: Writing complete application using array of objects

Due: Saturday, 18 December 2021, 12:00 AM

Instructions:

The application consist of four classes:

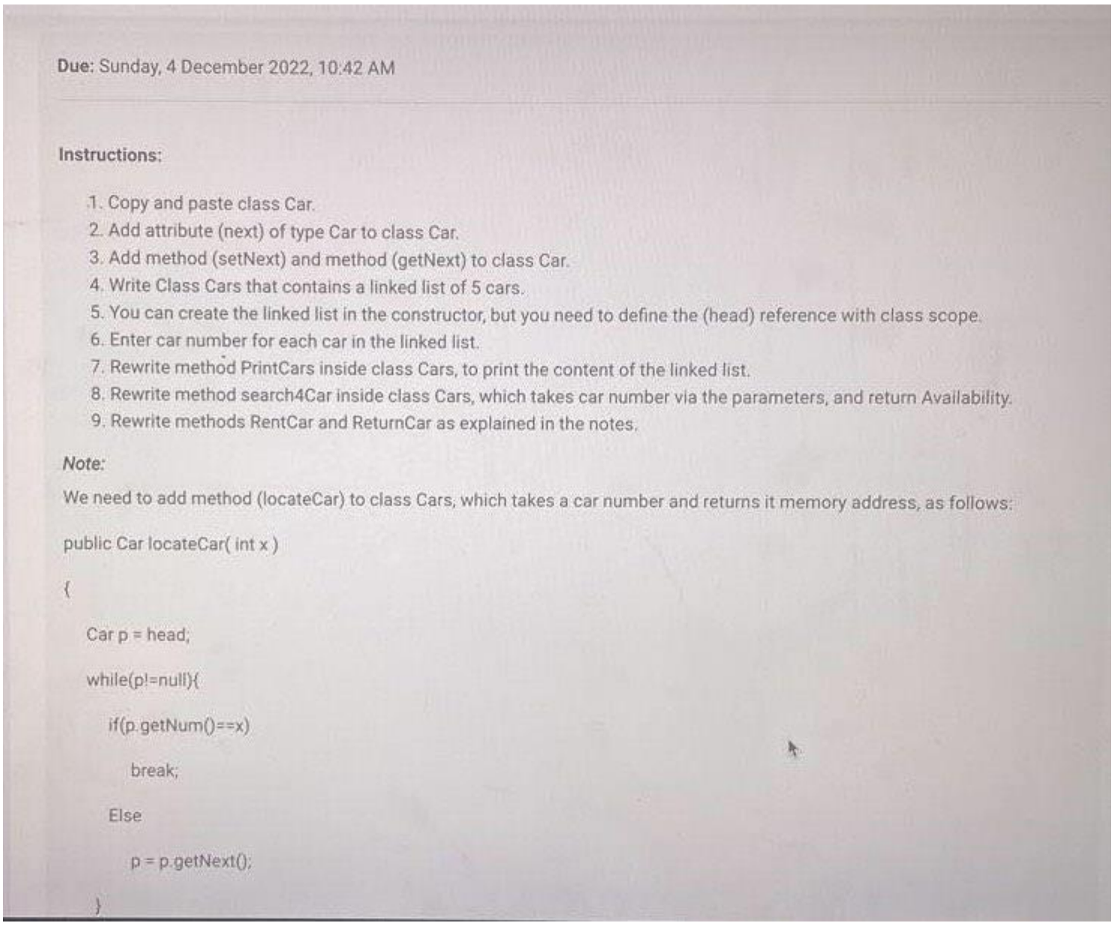
1. Program: This class contains method (Main). It is the default class.
2. Customer: You wrote this class in Lab 1. Just copy and paste it right after the Program class.
3. Car: You wrote this one too. Copy and paste it right after class Customer.
4. Cars: This class contains most of the code. It contains the array of cars, method for rent car, method for return car, method for searching car, method for printing cars, and method to display the main menu. You wrote part of this class in Lab 2. You can use it as well.

Description of the Application:

1. Display the main menu on the screen, which contains five options (Rent Car, Return Car, Search for Car, Print List of Cars, and exit).
2. The user of the application enter a number (1-5) to choose one of these five options.
3. If the user enter (1), then call method (RentCar). Return to the menu after completing of the renting task.
4. If the user enter (2), then call method (ReturnCar). Return to the main menu after completion of the returning task.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// Assignment 5
namespace Assignment4
{
    class Program
    {
        static void Main(string[] args)
        {

            Cars cs1 = new Cars();
            int chce = 0;
            chce = cs1.Menu();
            while (chce != 5)
            {
                switch (chce)
                {
                    case 1: cs1.Borrowcar(); break;
                    case 2: cs1.Returncar(); break;
                    case 3: cs1.Searchforcar(); break;
                    case 4: cs1.Printcars(); break;
                    default: Console.WriteLine("Invaild Choice. Try
Again"); break;
                }
                chce = cs1.Menu();
            }
        }
    }
    class Cars
    {
        private CarNumber[] c1;
        public Cars()
        {
            c1 = new CarNumber[3];
            for (int i = 0; i < 3; i++)
            {
                c1[i] = new CarNumber();
                c1[i].setNum(i + 100);
                c1[i].setcoustmer(null);
            }
        }
        public int Menu()
        {
            Console.WriteLine("\t my cars");
            Console.WriteLine("1. Borrow car");
            Console.WriteLine("2. Return car");
            Console.WriteLine("3. Search for car");
            Console.WriteLine("4. print cars");
            Console.WriteLine("5. Exit");
            Console.WriteLine("-----------------------");
            Console.WriteLine("Enter choice");
            int z = int.Parse(Console.ReadLine());
            return z;

        }
        public void Borrowcar()
        {
            Console.WriteLine(" Enter Car Name : ");
            string name = Console.ReadLine();
            Console.WriteLine(" Enter Car ID : ");
            int n = int.Parse(Console.ReadLine());
            Console.WriteLine(" Enter Car Number : ");
            int bn = int.Parse(Console.ReadLine());
            coustmer ca = new coustmer();
            ca.setName(name);
            ca.setID(n);
            bn = bn - 100;
            c1[bn].setcoustmer(ca);
        }
        public void Returncar()
        {
            Console.WriteLine(" Enter Car Number : ");
            int bn = int.Parse(Console.ReadLine());
            bn = bn - 100;
            c1[bn].setcoustmer(null);
        }
        public void Searchforcar()
        {
            Console.WriteLine(" Enter Car Number : ");
            int bn = int.Parse(Console.ReadLine());
            bn = bn - 100;
            Console.WriteLine(" Car No. : " + c1[bn].getNum());
            if (c1[bn].getcoustmer() == null)
                Console.WriteLine(" , Car is Available. ");
            else
            {
                Console.Write(" , " + c1[bn].getcoustmer().getName());
                Console.WriteLine(" , " + c1[bn].getcoustmer().getID());

            }
        }
        public void Printcars()
        {
            for (int i = 0; i < 3; i++)
            {
                Console.Write(" Car No. : " + c1[i].getNum());
                if (c1[i].getcoustmer() == null)
                    Console.WriteLine(" , Car is Available. ");
                else
                {
                    Console.Write(" , " + c1[i].getcoustmer().getName());
                    Console.WriteLine(" , " +
c1[i].getcoustmer().getID());
                }
            }
        }
    }
    class coustmer
    {
        private string name;
        private int ID;
        public void setName(string x)
        {
            name = x;
        }
        public string getName()
        {
            return name;
        }
        public void setID(int x)
        {
            ID = x;
        }
        public int getID()
        {
            return ID;
        }


    }
    class CarNumber
    {
        private int cNumb;
        private coustmer co;
        public void setNum(int x)
        {
            cNumb = x;

        }
        public void setcoustmer(coustmer x)
        {
            co = x;

        }
        public int getNum()
        {
            return cNumb;
        }
        public coustmer getcoustmer()
        {
            return co;
        }
    }
}
```

Assignment 6



Due: Sunday, 4 December 2022, 10:42 AM

Instructions:

1. Copy and paste class Car.
2. Add attribute (next) of type Car to class Car.
3. Add method (setNext) and method (getNext) to class Car.
4. Write Class Cars that contains a linked list of 5 cars.
5. You can create the linked list in the constructor, but you need to define the (head) reference with class scope.
6. Enter car number for each car in the linked list.
7. Rewrite method PrintCars inside class Cars, to print the content of the linked list.
8. Rewrite method search4Car inside class Cars, which takes car number via the parameters, and return Availability.
9. Rewrite methods RentCar and ReturnCar as explained in the notes.

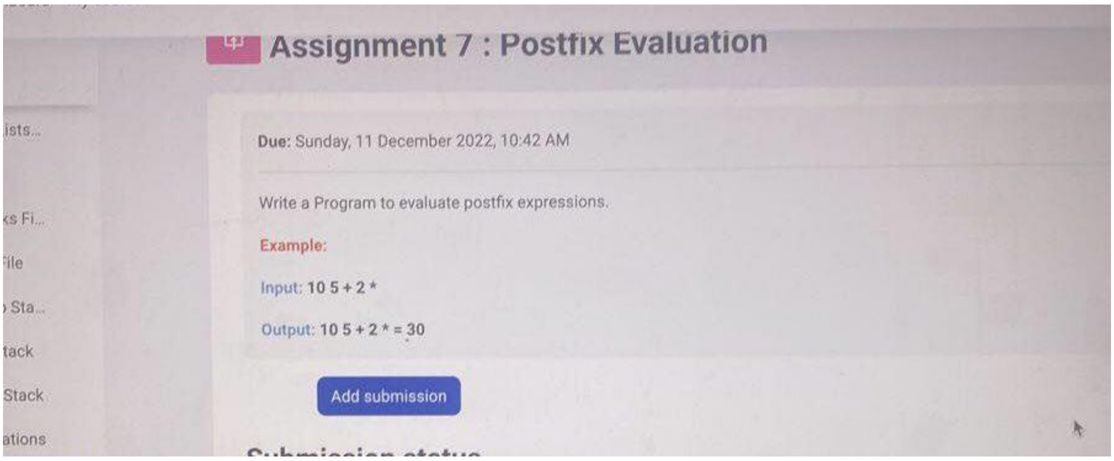
Note:

We need to add method (locateCar) to class Cars, which takes a car number and returns it memory address, as follows:

```
public Car locateCar( int x )

{

    Car p = head;

    while(p!=null){

       if(p.getNum()==x)

          break;

       Else

          p = p.getNext();

    }
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// Assignment 6
namespace Assignment_6
{
    class Program
    {
        static void Main(string[] args)
        {

            Cars cs1 = new Cars();
            int chce = 0;
            chce = cs1.Menu();
            while (chce != 5)
            {
                switch (chce)
                {
                    case 1: cs1.Borrowcar(); break;
                    case 2: cs1.Returncar(); break;
                    case 3: cs1.Searchforcar(); break;
                    case 4: cs1.Printcars(); break;
                    default: Console.WriteLine("Invaild Choice. Try
Again"); break;
                }
                chce = cs1.Menu();
            }
            Console.ReadKey();
        }
    }
    class Cars
    {
        private CarNumber c1;
        public Cars()
        {
            c1 = null;
            for (int i = 0; i < 3; i++)
            {
                CarNumber c = new CarNumber();
                c.setNum(i + 100);
                c.setcoustmer(null);
                c.setNext(c1);
                c1 = c;
            }
        }
        public int Menu()
        {
            Console.WriteLine("\t my cars");
            Console.WriteLine("1. Borrow car");
            Console.WriteLine("2. Return car");
            Console.WriteLine("3. Search for car");
            Console.WriteLine("4. print cars");
            Console.WriteLine("5. Exit");
            Console.WriteLine("-----------------------");
            Console.WriteLine("Enter choice");
            int z = int.Parse(Console.ReadLine());
            return z;

        }
        public void Borrowcar()
        {
            Console.WriteLine(" Enter Car Name : ");
            string name = Console.ReadLine();
            Console.WriteLine(" Enter Car ID : ");
            int n = int.Parse(Console.ReadLine());
            Console.WriteLine(" Enter Car Number : ");
            int bn = int.Parse(Console.ReadLine());
            coustmer ca = new coustmer();
            ca.setName(name);
            ca.setID(n);
            bn = bn - 100;
            //c1[bn].setcoustmer(ca);
        }
        public void Returncar()
        {
            Console.WriteLine(" Enter Car Number : ");
            int bn = int.Parse(Console.ReadLine());
            bn = bn - 100;
            //c1[bn].setcoustmer(null);
        }
        public void Searchforcar()
        {
            Console.WriteLine(" Enter Car Number : ");
            int bn = int.Parse(Console.ReadLine());
            bn = bn - 100;
        }
        public void Printcars()
        {
            CarNumber c = c1;
            while (c != null)
            {
                Console.Write(" Car No. : " + c.getNum());
                if (c.getcoustmer() == null)
                    Console.WriteLine(" , Car is Available. ");
                else
                {
                    Console.Write(" , " + c.getcoustmer().getName());
                    Console.WriteLine(" , " + c.getcoustmer().getID());
                }
                c = c.getNext();
            }
        }
    }
    class coustmer
    {
        private string name;
        private int ID;

        public void setName(string x)
        {
            name = x;
        }
        public string getName()
        {
            return name;
        }
        public void setID(int x)
        {
            ID = x;
        }
        public int getID()
        {
            return ID;
        }
    }
    class CarNumber
    {
        private int cNumb;
        private coustmer co;
        private CarNumber next;
        public void setNext(CarNumber x)
        {
            next = x;
        }
        public CarNumber getNext()
        {
            return next;
        }
        public void setNum(int x)
        {
            cNumb = x;

        }
        public void setcoustmer(coustmer x)
        {
            co = x;
        }
        public int getNum()
        {
            return cNumb;
        }
        public coustmer getcoustmer()
        {
            return co;
        }


    }
}
```

Assignment 7



Assignment 7 : Postfix Evaluation

Due: Sunday, 11 December 2022, 10:42 AM

Write a Program to evaluate postfix expressions.

Example:

Input: **10 5 + 2 ***

Output: **10 5 + 2 * = 30**

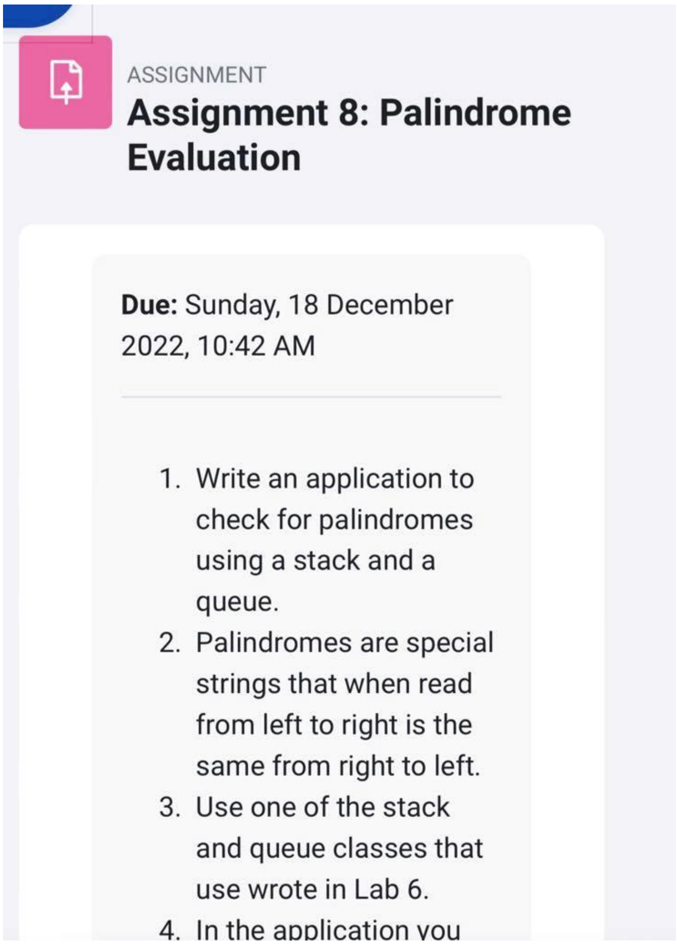
```
using System;
using System.Collections;


// Assignment 7
namespace Assignment_7
{
  using System;
  using System.Collections.Generic;


  class GFG
  {
    // Method to evaluate value of
    // a postfix expression
    public static int evaluatePostfix(string exp)
    {
      // create a stack
      Stack<int> stack = new Stack<int>();


      // Scan all characters one by one
      for (int i = 0; i < exp.Length; i++)
      {
        char c = exp[i];
```

```
if (c == ' ')
{
  continue;
}

// If the scanned character is an
// operand (number here),extract
// the number. Push it to the stack.
else if (char.IsDigit(c))
{
  int n = 0;

  // extract the characters and
  // store it in num
  while (char.IsDigit(c))
  {
    n = n * 10 + (int)(c - '0');
    i++;
    c = exp[i];
  }
  i--;

  // push the number in stack
  stack.Push(n);
}

// If the scanned character is
// an operator, pop two elements
// from stack apply the operator
else
```

```
    {
      int val1 = stack.Pop();
      int val2 = stack.Pop();

      switch (c)
      {
        case '+':
          stack.Push(val2 + val1);
          break;

        case '-':
          stack.Push(val2 - val1);
          break;

        case '/':
          stack.Push(val2 / val1);
          break;

        case '*':
          stack.Push(val2 * val1);
          break;
      }
    }
  }
  return stack.Pop();
}


public static void Main(string[] args)
{
  string exp = "10 5 + 2 *";
```

```
            Console.WriteLine("10 5 + 2 * = " + evaluatePostfix(exp));

            Console.ReadKey();

        }


    }

}


// This code is contributed by Shrikant13
```

Assignment 8



ASSIGNMENT

Assignment 8: Palindrome Evaluation

Due: Sunday, 18 December 2022, 10:42 AM

1. Write an application to check for palindromes using a stack and a queue.
2. Palindromes are special strings that when read from left to right is the same from right to left.
3. Use one of the stack and queue classes that use wrote in Lab 6.
4. In the application you

```
using System;

using System.Text;

// Assignment 8

class Stack

{

  public int size;

  public int top;

  public char[] a;


  // function to check if stack is empty

  public Boolean isEmpty()

  {

    return (top < 0);
```

```
}

public Stack(int n)
{
  top = -1;
  size = n;
  a = new char[size];
}

// function to push element in Stack
public Boolean push(char x)
{
  if (top >= size)
  {
    Console.WriteLine("Stack Overflow");
    return false;
  }
  else
  {
    a[++top] = x;
    return true;
  }
}

// function to pop element from stack
public char pop()
{
  if (top < 0)
  {
    Console.WriteLine("Stack Underflow");
    return ' ';
```

```
        }
        else
        {
            char x = a[top--];
            return x;
        }
    }
}

class GFG
{
    // function to reverse the string
    public static void reverse(StringBuilder str)
    {
        // Create a stack of capacity
        // equal to length of string
        int n = str.Length;
        Stack obj = new Stack(n);

        // Push all characters of string
        // to stack
        int i;
        for (i = 0; i < n; i++)
            obj.push(str[i]);

        // Pop all characters of string
        // and put them back to str
        for (i = 0; i < n; i++)
        {
            char ch = obj.pop();
            str[i] = ch;
        }
    }

    // Driver code
    public static void Main(String[] args)
    {
        // create a new string
        StringBuilder s = new StringBuilder("12321");
        StringBuilder s1 = new StringBuilder("MAM");
        StringBuilder s2 = new StringBuilder("12 12 21 21");
        StringBuilder s3 = new StringBuilder("ABLE WAS I ERE I SAW ELBA");

        // call reverse method
        reverse(s);

        // print the reversed string
        Console.WriteLine("Some examples on palindromes are: " + s + " - " + s1 + " - " + s2 +
" - " + s3);
    }
}

// This code is contributed by Rajput-Ji
```

Project Hotel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment
{
  class Program
  {
    static void Main(string[] args)
    {

      Rooms room = new Rooms();

      while (true)
      {
        Console.WriteLine("My Hotel\n ");
        Console.WriteLine("Main Menu\n ");
        Console.WriteLine("1. Reserve Room");
        Console.WriteLine("2. Return Room");
        Console.WriteLine("3. Search 4 Room");
        Console.WriteLine("4. Print List of Rooms");
        Console.WriteLine("5. Exit");
        Console.WriteLine(" Enter your choice :");
        string input = Console.ReadLine();

        switch (input)
        {
          case "1":
}                   room.RentRoom();
                    break;
                case "2":
                    room.ReturnRoom();
                    break;
                case "3":
                    Console.WriteLine("Enter a room number to search for:");
                    int roomNumberToSearch = int.Parse(Console.ReadLine());
                    bool roomFound = room.Search4Room(roomNumberToSearch);
                    if (roomFound)
                    {
                        Console.WriteLine("Room is Available");
                    }
                    else
                    {
                        Console.WriteLine("Room is not Available");
                    }
                    break;
                case "4":
                    room.PrintRooms();
                    break;
                case "5":
                    return;
                default:
                    Console.WriteLine("Invalid input. Please try again.");
                    break;
            }
            Console.WriteLine();
        }
    }
}
```

```
class Room
{
  private int RoomNumber;
  private Room next;

  public int GetRoomNumber()
  {
    return RoomNumber;
  }
  public void SetRoomNumber(int room)
  {
    RoomNumber = room;
  }

  public Room GetNext()
  {
    return next;
  }
  public void SetNext(Room nextRoom)
  {
    next = nextRoom;
  }
}
class Rooms
{
  private Room head;

  private Room[] rooms;

  public Rooms()
```

```
{
    head = null;

    for (int i = 0; i < 5; i++)
    {
        Room newRoom = new Room();
        newRoom.SetRoomNumber(i + 1);
        newRoom.SetNext(head);
        head = newRoom;
    }

    rooms = new Room[5];
    for (int i = 0; i < 5; i++)
    {
        rooms[i] = LocateRoom(i + 1);
    }
}
public void RentRoom()
{
    Console.WriteLine("My Hotel");
    Console.WriteLine("Rent Room");
    Console.Write("Enter Room No: ");
    int roomNumber = int.Parse(Console.ReadLine());

    Room roomToRent = null;
    for (int i = 0; i < rooms.Length; i++)
    {
        if (rooms[i].GetRoomNumber() == roomNumber)
        {
            roomToRent = rooms[i];
            break;
        }
    }

    if (roomToRent != null)
    {

        Console.WriteLine("Room Number: {0}", roomToRent.GetRoomNumber());
        Console.Write("Correct? (Y/N) ");
        string confirmation = Console.ReadLine();
        if (confirmation.Equals("Y", StringComparison.OrdinalIgnoreCase))
        {
            Console.WriteLine("Room is already rented by a customer.");
        }
    }
    else
    {
        Console.WriteLine("Room not found.");
    }
}
public void ReturnRoom()
{
    Console.WriteLine("My Hotel");
    Console.WriteLine("Return Room");
    Console.Write("Enter Room No: ");
    int roomNumber = int.Parse(Console.ReadLine());

    Room roomToReturn = null;
    for (int i = 0; i < rooms.Length; i++)
    {
        if (rooms[i].GetRoomNumber() == roomNumber)
        {
            roomToReturn = rooms[i];
            break;
        }
    }

    if (roomToReturn != null)
    {
        Console.WriteLine("Room Number: {0}", roomToReturn.GetRoomNumber());
        Console.Write("Correct? (Y/N) ");
        string confirmation = Console.ReadLine();
        if (confirmation.Equals("Y", StringComparison.OrdinalIgnoreCase))
        {
            Console.WriteLine("No customer assigned to this room. Cannot return.");
        }
    }
    else
    {
        Console.WriteLine("Room not found.");
    }
}

public void PrintRooms()
{
    for (int i = 0; i < 5; i++)
    {
        Console.WriteLine("Room Number: {0}", rooms[i].GetRoomNumber());
    }
}

public bool Search4Room(int roomNumber)
{
```

```
            for (int i = 0; i < 3; i++)
            {
                if (rooms[i].GetRoomNumber() == roomNumber)
                {
                    return true;
                }
            }
            return false;
        }
        public Room LocateRoom(int roomNumber)
        {
            Room currentRoom = head;

            while (currentRoom != null)
            {
                if (currentRoom.GetRoomNumber() == roomNumber)
                    break;
                else
                    currentRoom = currentRoom.GetNext();
            }

            return currentRoom;
        }


    }

}
```

Project Hotel

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace myhotell
{
   class Program
   {
      static void Main(string[] args)
      {
         Customer customer1 = new Customer();
         Customer customer2 = new Customer();
         Rooms room = new Rooms();

         customer1.SetCustomerName("khaled");
         customer1.SetCustomerID(15448);

         customer2.SetCustomerName("mohammad");
         customer2.SetCustomerID(54878);

         Room room1 = new Room();
         Room room2 = new Room();

         /*Console.WriteLine("Customer1 Name: {0}, Customer1 ID: {1}", customer1.GetCustomerName(), customer1.GetCustomerID());
         Console.WriteLine("Customer2 Name: {0}, Customer2 ID: {1}", customer2.GetCustomerName(), customer2.GetCustomerID());
         Console.WriteLine("RoomNumber: {0}, Custoername : {1}", 505, "khaled yousef");
Console.WriteLine("RoomNumber: {0}, Customername : {1}", 909, "mohammad hashem ");*/


while (true)

{

    Console.WriteLine("My Hotel\n ");

    Console.WriteLine("Main Menu\n ");

    Console.WriteLine("1. Reserve Room\r\n");

    Console.WriteLine("2. Return Room");

    Console.WriteLine("3. Search 4 Room");

    Console.WriteLine("4. Print List of Rooms");

    Console.WriteLine("5. Exit");

    Console.WriteLine(" Enter your choice :");

    string input = Console.ReadLine();


    switch (input)

    {

        case "1":

            room.RentRoom();

            break;

        case "2":

            room.ReturnRoom();

            break;

        case "3":

            Console.WriteLine("Enter a room number to search for:");

            int roomNumberToSearch = int.Parse(Console.ReadLine());

            bool roomFound = room.Search4Room(roomNumberToSearch);

            if (roomFound)

            {

                Console.WriteLine("Room is Available");

            }
```

```
                    else
                    {
                        Console.WriteLine("Room is not Available");
                    }
                    break;
                case "4":
                    room.PrintRooms();
                    break;
                case "5":
                    return;
                default:
                    Console.WriteLine("Invalid input. Please try again.");
                    break;
            }
            Console.WriteLine();
        }
    }
}

class Customer
{
    private string customerName;
    private int customerID;
    public string GetCustomerName()
    {
        return customerName;
    }

    public void SetCustomerName(string name)
    {
        customerName = name;
    }

    public int GetCustomerID()
    {
        return customerID;
    }

    public void SetCustomerID(int id)
    {
        customerID = id;
    }
}
class Room
{
    private int RoomNumber;
    private Customer customer;

    public int GetRoomNumber()
    {
        return RoomNumber;
    }
    public void SetRoomnumber(int room)
    {
        RoomNumber = room;
    }
    public Customer Getcustomer1()
    {
        return customer;
    }
    public void Setcustomer(Customer customer1)
    {
        customer = customer1;
    }
}


class Rooms
{
    private Room[] rooms;
    public Rooms()
    {
        rooms = new Room[3];
        for (int i = 0; i < 3; i++)
        {
            rooms[i] = new Room();
        }
        rooms[0].SetRoomnumber(5);
        rooms[1].SetRoomnumber(6);
        rooms[2].SetRoomnumber(7);
    }
    public void RentRoom()
    {
        Console.WriteLine("My Hotel");
        Console.WriteLine("Rent Room");
        Console.Write("Enter Customer ID: ");
        int customerID = int.Parse(Console.ReadLine());
        Console.Write("Enter Room No: ");
        int roomNumber = int.Parse(Console.ReadLine());

        Room roomToRent = null;
        for (int i = 0; i < rooms.Length; i++)
        {
```

```
    if (rooms[i].GetRoomNumber() == roomNumber)
    {
        roomToRent = rooms[i];
        break;
    }
}

if (roomToRent != null)
{
    Customer customer = roomToRent.Getcustomer1();
    if (customer != null)
    {
        Console.WriteLine("Customer Name: {0}", customer.GetCustomerName());
    }
    else
    {
        Console.WriteLine("No customer assigned to this room.");
    }

    Console.WriteLine("Room Number: {0}", roomToRent.GetRoomNumber());
    Console.Write("Correct? (Y/N) ");
    string confirmation = Console.ReadLine();
    if (confirmation.Equals("Y", StringComparison.OrdinalIgnoreCase))
    {

        if (customer == null)
        {
            Customer newCustomer = new Customer();
            newCustomer.SetCustomerID(customerID);

            newCustomer.SetCustomerName("Sample Customer");
```

```
                roomToRent.Setcustomer(newCustomer);
                Console.WriteLine("Room rented successfully!");
            }
            else
            {
                Console.WriteLine("Room is already rented by a customer.");
            }
        }
    }
    else
    {
        Console.WriteLine("Room not found.");
    }
}
public void ReturnRoom()
{
    Console.WriteLine("My Hotel");
    Console.WriteLine("Return Room");
    Console.Write("Enter Room No: ");
    int roomNumber = int.Parse(Console.ReadLine());

    Room roomToReturn = null;
    for (int i = 0; i < rooms.Length; i++)
    {
        if (rooms[i].GetRoomNumber() == roomNumber)
        {
            roomToReturn = rooms[i];
            break;
        }
    }
```

```
if (roomToReturn != null)
{
    Customer customer = roomToReturn.Getcustomer1();
    if (customer != null)
    {
        Console.WriteLine("Customer Name: {0}", customer.GetCustomerName());
    }
    else
    {
        Console.WriteLine("No customer assigned to this room.");
    }

    Console.WriteLine("Room Number: {0}", roomToReturn.GetRoomNumber());
    Console.Write("Correct? (Y/N) ");
    string confirmation = Console.ReadLine();
    if (confirmation.Equals("Y", StringComparison.OrdinalIgnoreCase))
    {

        if (customer != null)
        {
            roomToReturn.Setcustomer(null);
            Console.WriteLine("Room returned successfully!");
        }
        else
        {
            Console.WriteLine("No customer assigned to this room. Cannot return.");
        }
    }
}
else
{
        Console.WriteLine("Room not found.");
    }
}

public void PrintRooms()
{
    for (int i = 0; i < 3; i++)
    {
        Console.WriteLine("Room Number: {0}", rooms[i].GetRoomNumber());
        if (rooms[i].Getcustomer1() != null)
        {
            string customername = rooms[i].Getcustomer1().GetCustomerName();
            int customerid = rooms[i].Getcustomer1().GetCustomerID();
            Console.WriteLine("Customer Name: {0}", customername);
            Console.WriteLine("Customer ID: {0}", customerid);
        }
    }
}

public bool Search4Room(int roomNumber)
{
    for (int i = 0; i < 3; i++)
    {
        if (rooms[i].GetRoomNumber() == roomNumber)
        {
            if (rooms[i].Getcustomer1() != null)
            {
                string customername = rooms[i].Getcustomer1().GetCustomerName();
                int customerid = rooms[i].Getcustomer1().GetCustomerID();
                Console.WriteLine("Customer Name: {0}", customername);
                Console.WriteLine("Customer ID: {0}", customerid);
```

```
                    }
                    return true;
                }
            }
            return false;
        }
    }
}
```