**T123: ICT272 Web Design and Development**

**Tutorial 3**

**Topic: Object-oriented programming**

**Submission: Submit to Moodle link (Week3)**

**Exercise 1 – Answer the following questions:**

* Explain how objects are related to each other in an app which lets user purchase products online.

ANSWER

A software that enables online product purchases by users using API connects the items to one another (Application Programming Interface). Without knowing how they are implemented; you can use API to assist your objects in communicating with one another. You can save time and money by doing this. As you create new tools and objects, APIs allow your program flexibility.

For instance, as soon as you click Add to Cart, the API alerts the website where you added the product to the cart, causing the website to add the product and update your cart. This indicates that the API sends the user's response to an object as well as returning the user's response from the object.

* What is the difference between constructors and methods in classes?

ANSWER

* + The constructor initializes an objects of that class in which they are declared, whereas method tells about the functionality of an object.
  + The constructor does not return any value in the class program because any return is not mentioned before the constructor’s name, while the method may or may not return the value, because the return type may or may not be mentioned before the method or function name.
  + Constructors are invoked automatically when the object is created in the class program, while methods are executed by the object of that class explicitly or forcefully.

**Exercise 2: Create an object from a class with a single field.**

* **Write a program in C# to create a coffee Filter Class (C Filter) with a size field.**
* **create an object called "CF\_001" with size 31 and use it to print the value of size.**

**ANSWER**

C# that creates a Coffee Filter class and an object "CF\_001" with size 31:

using System;

class CoffeeFilter {

public int Size;

public CoffeeFilter(int size) {

this.Size = size;

}

}

class Program {

static void Main(string [] args) {

CoffeeFilter CF\_001 = new CoffeeFilter(31);

Console.WriteLine("Size of CF\_001: " + CF\_001.Size);

}

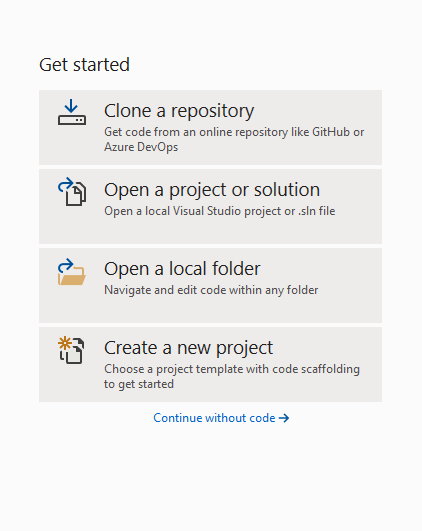
}

In this code, we create a **CoffeeFilter** class with a public **Size** field, and a constructor that takes a **size** argument and assigns it to the **Size** field. Then, in the **Main** method, we create an object **CF\_001** of type **CoffeeFilter** with a size of 31 and print the value of its **Size** field using **Console.WriteLine()**.

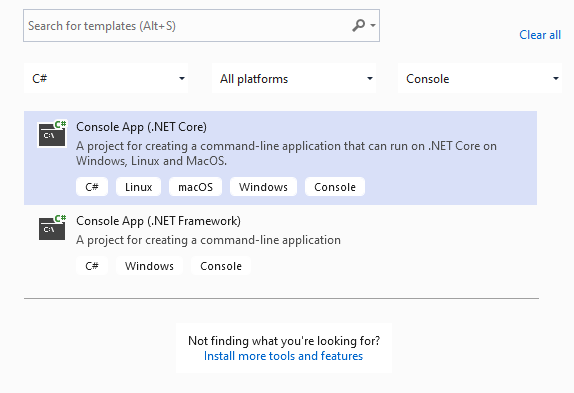
When you run this program, it should output:

Size of CF\_001: 31

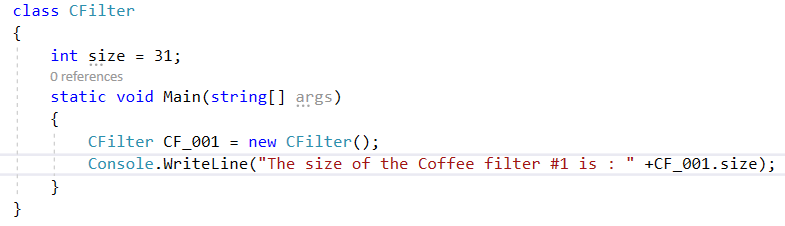
1. **Start your visual studio 2019.**
2. **Click on Create a new project.**



1. Search for console in search box and select “Console App (.Net Core)



1. Name you project “Tutorial4Ex2” and click on Create.
2. Inside the main function add the code below



1. Click on start to run the application and see the output.

**What to submit:**

1. The source code of Program file (Main function)
2. The output of the program

ANSWER:

namespace Tutorial4Ex2

{

internal class Program

{

Class CFilter

{

int size = 31;

static void Main(string[] args)

{

CFilter CF\_001 = new CFilter();

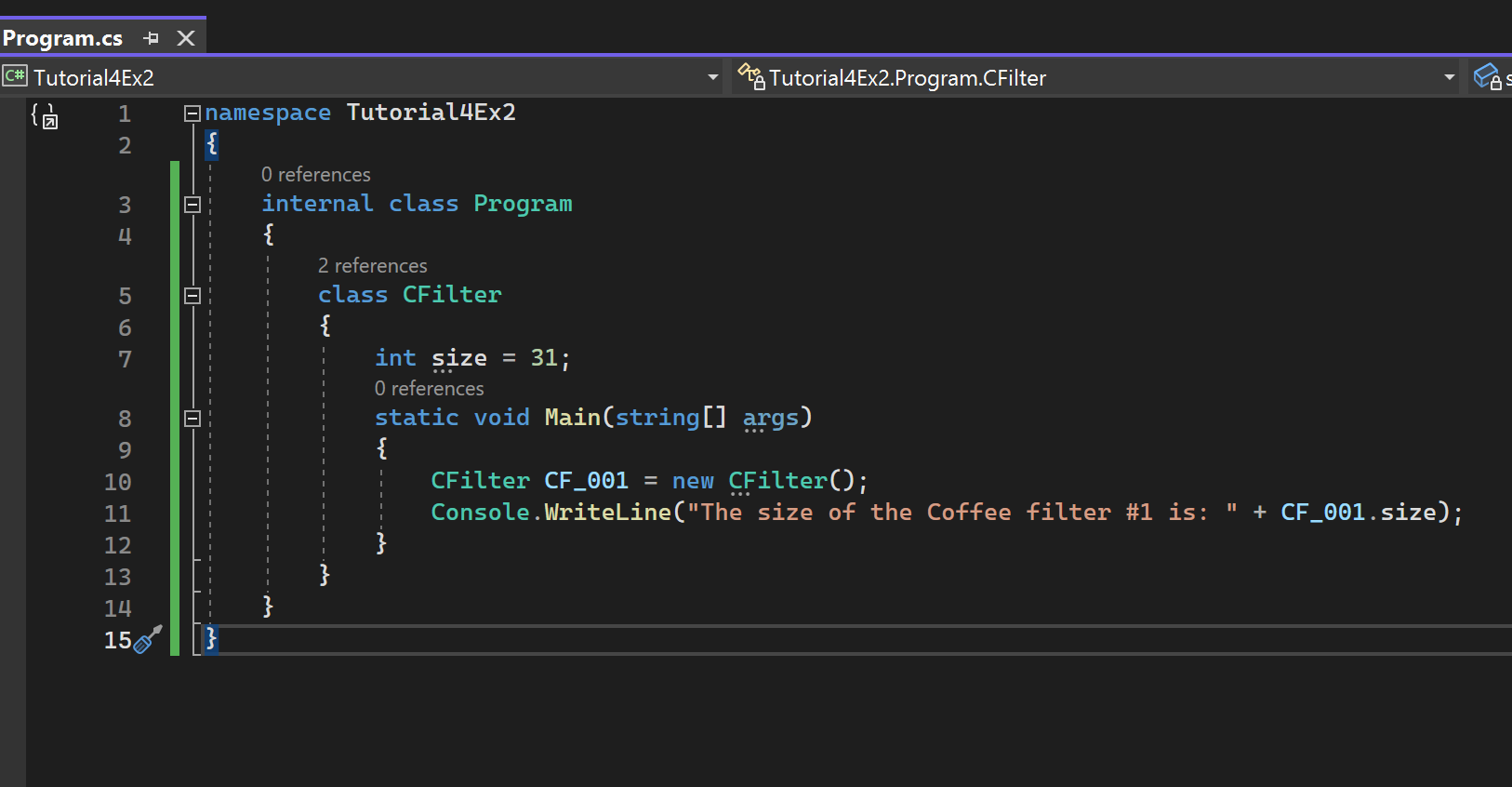
Console.WriteLine("The size of the Coffee filter #1 is: " + CF\_001.size);

}

}

}

}



Graphical user interface, text, website

Description automatically generated

**Exercise 3: Create a class and use it with another class using access modifier.**

* **Create a class called Customer with 3 properties (id, name, and email) and a constructor.**
* **Use it to create 2 objects (Cust1 and Cust2) in the Program class.**

**ANSWER**

using System;

namespace Tutorial4Ex3

{

class Customer

{

public int id;

public string name;

public string email;

public Customer(int id, string name, string email)

{

this.id = id;

this.name = name;

this.email = email;

}

}

class Program

{

static void Main(string[] args)

{

// Creating two instances of the Customer class

Customer cust1 = new Customer(1, "John Smith", "john.smith@example.com");

Customer cust2 = new Customer(2, "Jane Doe", "jane.doe@example.com");

// Printing out the properties of the Customer objects

Console.WriteLine($"Customer 1 - ID: {cust1.id}, Name: {cust1.name}, Email: {cust1.email}");

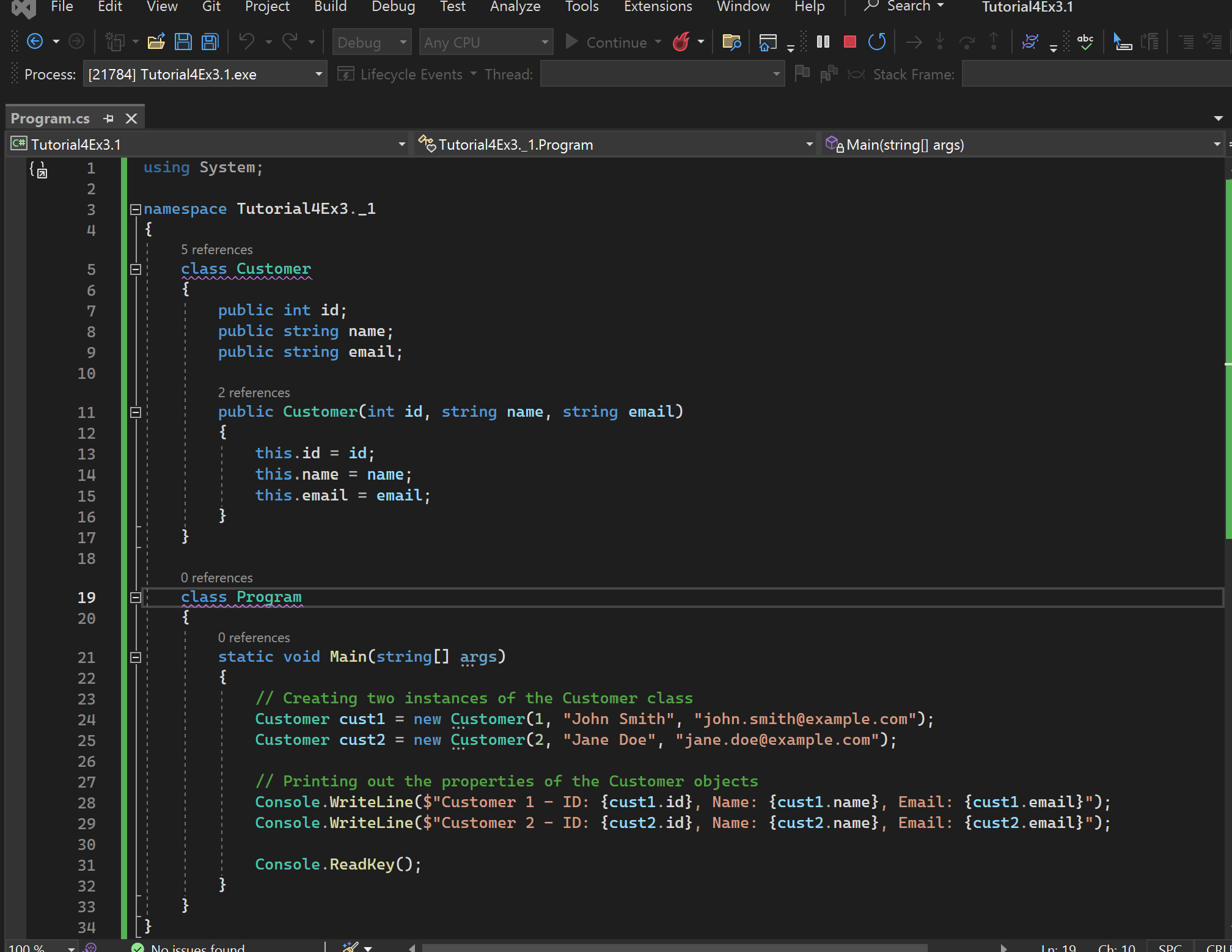
Console.WriteLine($"Customer 2 - ID: {cust2.id}, Name: {cust2.name}, Email: {cust2.email}");

Console.ReadKey();

}

}

}

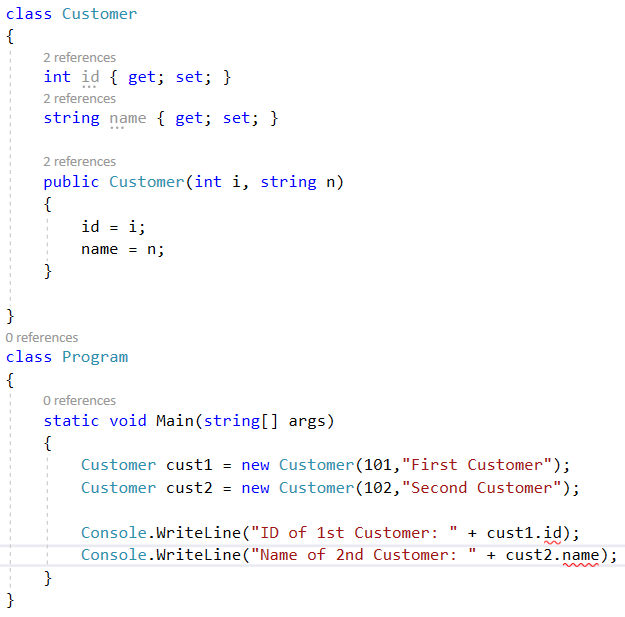


Text

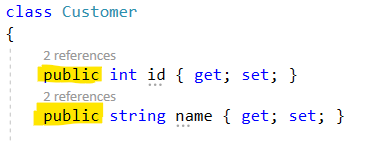
Description automatically generated with medium confidence

1. **Start your visual studio 2019.**
2. **Click on Create a new project.**
3. **Search for console in search box and select “Console App (.Net Core)**
4. **Name you project “Tutorial4Ex3” and click on Create.**
5. **Create the below 2 classes in the namespace:**

**Class Customer and class Program**



1. Click on start to run the application and **observe the error which you receive.**
2. Now add ‘public’ keyword to the 2 properties in Customer class as per below.



1. Click on start to run the application and see the output.

**What to submit:**

1. The source code of Program file (Main function)
2. The output of the program

ANSWER**:**

using System;

namespace Tutorial4Ex3

{

class Customer

{

public int id { get; set; }

public string name { get; set; }

public Customer(int i, string n)

{

id = i;

name = n;

}

}

class Program

{

static void Main(string[] args)

{

Customer cust1 = new Customer(101, "First Customer");

Customer cust2 = new Customer(102, "Second Customer");

Console.WriteLine("ID of 1st Customer: " + cust1.id);

Console.WriteLine("ID of 2nd Customer: " + cust2.id);

Console.ReadKey();

}

}

}

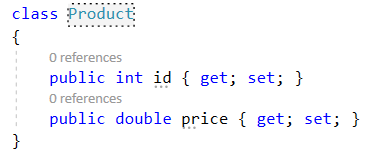
**Graphical user interface, text

Description automatically generated**

**Exercise 4**:

* **Add another class called Product to the above namespace**
* **Create a list in the Customer class to hold multiple products.**

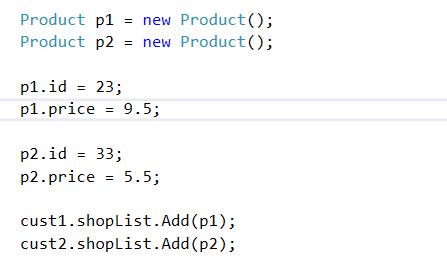
1. Create the Product class as below in the namespace:

****

1. Include the below namespace at the top to use Lists.

****

1. ****Add the below code to the Customer Class
2. Create 2 Product objects and them to Cust1’s shopping list by adding below code to Main method in Program class.



**What to submit:**

1. The source code of Program file (Main function)

using System;

using System.Collections.Generic;

namespace Tutorial4Ex3

{

class Product

{

public int id { get; set; }

public double price { get; set; }

}

class Customer

{

public int id { get; set; }

public string name { get; set; }

public List<Product> shopList = new List<Product>();

public Customer(int i, string n)

{

id = i;

name = n;

}

}

class Program

{

static void Main(string[] args)

{

Customer cust1 = new Customer(101, "First Customer");

Customer cust2 = new Customer(102, "Second Customer");

Product p1 = new Product();

Product p2 = new Product();

p1.id = 23;

p1.price = 9.5;

p2.id = 33;

p2.price = 5.5;

cust1.shopList.Add(p1);

cust1.shopList.Add(p2);

Console.WriteLine("ID of 1st Customer: " + cust1.id);

Console.WriteLine("Shopping List of 1st Customer:");

foreach (Product item in cust1.shopList)

{

Console.WriteLine("Product ID: " + item.id + ", Price: " + item.price);

}

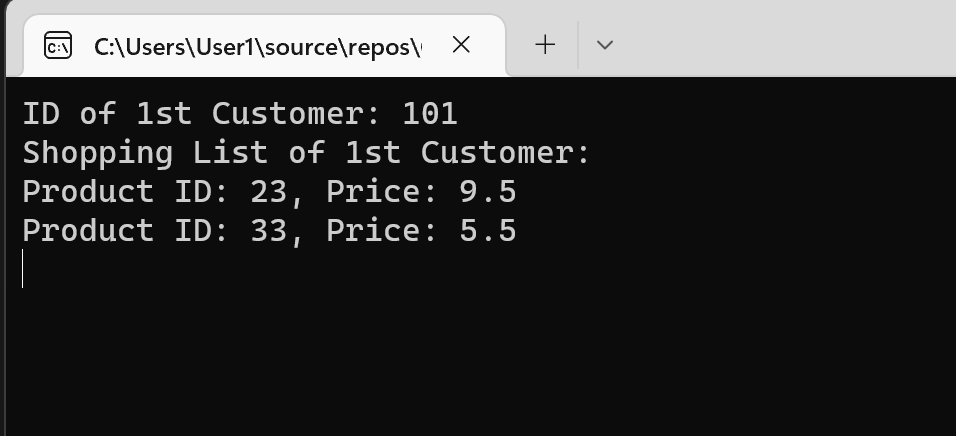
Console.ReadKey();

}

}

}

1. The output of the program



**END of Tutorial**