Explanation:

There are 7 classes in this application as follows.

1. Product class
2. DiscountedProduct class inherits Product class.
3. Customer class
4. PremiumCustomer class inherits Customer class.
5. Order class
6. Store class
7. Program class

1.Product class:

I have Created a Product class that has the following data members:

Attributes:

Number: A universal static variable that is increased whenever a new product is introduced
id: Unique identifier for the product.
name: Name of the product.
price: Price of the product.

constructors:

default constructor: It increases the static variable no by 1 to initialise the product's ID.
Parameterized constructor: accepts the product's name and price as two parameters, then initialises the product's characteristics.

Methods:

Each attribute's setter and getter methods. showProduct() displays the product's information.
calculate_discount() returns 0 for each product's discount.

2.DiscountedProduct class:

I have made the DiscountedProduct class to inherit from the Product class and include the following data elements in it:

Attributes:

Discount_rate: discount rate on product

constructors:

Call the base class constructor in the default constructor.
Constructor with three parameters: Name, Price, and Discount Rate of Product. Passes two parameters to constructor of base class to initialise properties.

Methods:

Discount approach using a setter and getter.

Calculate_discount(): overrides the base class's method to compute a product's discount depending on price.

showProduct(): overrides the base class's method to display discounted products.

3.Customer class:

I have Created a customer class that has the following data members:

Attributes:

No, a static variable that is utilised by all customers and is increased whenever a new client joins the group.

id: The client's distinctive identity. identifier: The client's name.

Customer's email address is provided here.

constructors:

The customer's ID is initialised by incrementing the static variable no by 1 in the default constructor.

Constructor with parameters initialises customer characteristics using name and email as two parameters.

Methods:

For each attribute, there are setter and getter methods. showCustomer(): This function provides customer information._discount() gives the buyer a discount of 0.

4.PremiumCustomer class:

I have Created a PremiumCustomer class which inherits Customer class and has the following data members:

Attributes:

Discount: premium customer discount

constructors:

Default invokes the base class's constructor.

The premium customer's name, email address, and discount rate are sent as three parameters to the constructor of the base class, which initialises the properties.

Methods:

showCustomer(): override basic class function to showcase premium customer receivesTo return the discount for the premium customer, override the base class function _discount().

5.Order class:

I have made a class called Order that symbolises a customer's order. It should possess the following qualities:

No, a static variable called id is utilised by every client and is incremented whenever a new one signs up. a special code for the purchase.
customer: The person who made the purchase.
products: A list of the items the consumer has ordered. total: The order's entire cost.
constructors:
The order's ID is initialised by incrementing the static variable no by 1 in the default constructor.
A constructor with parameters accepts two Customer objects, a list of items, and is used to initialise order properties.

Methods:

Each attribute's setter and getter methods.
Calculate the total price using the calculateTotal() function depending on the product, the discounted product, the customer, or the premium customer.
display order details with showOrder()

6.Store class:

I have made a class called "Store" that controls the items, clients, and orders. It should be capable of the following:

attributes:

clients: A list of clients. products: A collection of items.
A list of orders is an order.

constructors:

The standard constructor initialises every list.

Methods:

Add a new product to the list of products using the addProduct() function. The addCustomer() function includes a new customer in the list of customers. This function, addOrder(), adds a new order to the list of orders.
Display information about all goods, including discounted products, with fetchproducts().
Fetchorders(): Show information about all orders FetchCustomers(): Show information about all customers, including premium customers
discover a product using the product id and provide an object of the product if it is found; else, return null.
locate a customer using the customer id and return the object of that customer if it is found; else, return null.
locate an order using the order id and return an object of the identified order; otherwise, return null.

7.Program class:

There is a main method in this class. Create an object of the Store class using this method. The user is then given a menu with options to add new items, customers, orders, see all products and customers, view a specific order, and depart.

By calling the proper method from a certain class, the programme delivers functionality based on menu choices.

Code:

```
using System;
namespace Asgn3
{
    class Program
    {
        static void Main(string[] args)
        {
            Store s = new Store();
            int c;
            do
            {
                Console.Write("Select any one option from below \n");
                Console.WriteLine("1: Add Product\n2: Add Customer\n3: Add Order\n4:
Show All Products\n5: Show All Customers\n6: Show Order\n7: Exit\n");
                Console.Write("Enter Your Choice: ");
                c = int.Parse(Console.ReadLine());
                switch (c)
                {
                    case 1:
                        Console.Write("Enter Product Name:");
                        string pname = Console.ReadLine();
                        Console.Write("Enter Product Price:");
                        double price = double.Parse(Console.ReadLine());
                        Console.Write("Is it discounted product? (y for Yes and n for No): ");
                        string ans = Console.ReadLine().ToLower();
                        if (ans.Equals("n"))
                            s.addProduct(new Product(pname, price));
                        else
                        {
                            Console.Write("Enter Discount Rate(%):");
                            double dis = double.Parse(Console.ReadLine());
                            s.addProduct(new DiscountedProduct(pname, price, dis));
                        }
                        Console.WriteLine("Product Added");
                        break;
                    case 2:
                        Console.Write("Enter Customer Name:");
                        string cname = Console.ReadLine();
                        Console.Write("Enter Customer Email:");
                        string em = Console.ReadLine();
                        Console.Write("Is he/she Premium Customer? (y for Yes and n for No): ");
                        string ans1 = Console.ReadLine().ToLower();
                        if (ans1.Equals("n"))
                            s.addCustomer(new Customer(cname, em));
                        else
                        {
                            Console.Write("Enter Premium Discount(%):");
                            double dis = double.Parse(Console.ReadLine());
                            s.addCustomer(new PremiumCustomer(cname, em, dis));
                        }
                        Console.WriteLine("Customer Added");
                        break;
                    case 3:
                        Console.WriteLine("Customers:");
                        s.fetchcustomers();
                        Customer customer = null;
                        do
                        {
                            Console.Write("Enter Customer Id : ");
                            int Customerid = int.Parse(Console.ReadLine());
                            if (Customerid == 0)
                                break;
                            customer = s.getCustomerById(Customerid);
                            if (customer == null)
                            {
                                Console.WriteLine("Customer not Found");
                            }
                        } while (customer == null);
                        Console.WriteLine("Products:");
                        s.fetchproducts();
                        List<Product> prod = new List<Product>();
                        int Productid = 0;
                        do
                        {
                            Console.Write("Enter Product Id : ");
                            Productid = int.Parse(Console.ReadLine());
                            if (Productid == 0)
                                break;
                            Product p = s.getProductById(Productid);
                            if (p == null)
                            {
                                Console.WriteLine("Product is not found");
                            }
                            else
                            {
                                prod.Add(p);
                                Console.WriteLine("Product is Added to your Order");
                            }
                        }
                        while (Productid != 0);
                        if (customer != null)
                        {
                            Order oo = new Order(customer, prod);
                            oo.calculateTotal();
                            s.addOrder(oo);
                            Console.WriteLine("Order Added");
                        }
                        break;
                    case 4:
                        Console.WriteLine("Products:");
                        s.fetchproducts();
                        break;
                    case 5:
                        Console.WriteLine("Customers:");
                        s.fetchcustomers();
                        break;
                    case 6:
                        Console.Write("Enter Order Id: ");
                        int oid = int.Parse(Console.ReadLine());
                        Order o = s.getOrderById(oid);
                        if (o == null)
                        {
                            Console.WriteLine("Order not found");
                        }
                        else
                        {
                            o.showOrder();
                        }
```

```
                    break;
                case 7:
                    Console.WriteLine("Thank You, Visit again our Store");
                    break;
                default:
                    Console.WriteLine("Invalid Choice");
                    break;
            }
        }
        while (c != 7);
        Console.ReadKey();
    }
}
class Product
{
    protected static int Number = 0;
    protected int id;
    protected string name;
    protected double price;
    public Product()
    {
        Number++;
        this.id = Number;
    }
    public Product(string name, double price)
    {
        Number++;
        this.id = Number;
        this.name = name;
        this.price = price;
    }
    public int getId()
    {
        return this.id;
    }
    public void setName(string name)
    {
        this.name = name;
    }
    public string getName()
    {
        return this.name;
    }
```

```
        public void setPrice(double price)
        {
            this.price = price;
        }
        public double getPrice()
        {
            return price;
        }
        public void showProduct()
        {
            Console.WriteLine("Id: {0}, Name: {1}, Price: {2}", getId(), getName(),
getPrice());
        }
        public double calculate_discount()
        {
            return 0;
        }
    }
    class DiscountedProduct : Product
    {
        private double discount_rate;
        public DiscountedProduct() : base()
        {
        }
        public DiscountedProduct(string name, double price, double rate) : base(name,
price)
        {
            discount_rate = rate;
        }
        public void setDiscount(double rate)
        {
            discount_rate = rate;
        }
        public double getDiscount()
        {
            return discount_rate;
        }
        new public double calculate_discount()
        {
            return price * discount_rate / 100;
        }
        new public void showProduct()
        {
            Console.WriteLine("Id: {0}, Name: {1}, Price: {2}, Discount: {3}, Net Price:
{4}", getId(), getName(), getPrice(), calculate_discount(), getPrice() -
calculate_discount());
        }
    }
    class Customer
    {
        protected static int Number = 0;
        protected int id;
        protected string name;
        protected string email;
        public Customer()
        {
            Number++;
            this.id = Number;
        }
        public Customer(string name, string email)
        {
            Number++;
            this.id = Number;
            this.name = name;
            this.email = email;
        }
        public int getId()
        {
            return this.id;
        }
        public void setName(string name)
        {
            this.name = name;
        }
        public string getName()
        {
            return this.name;
        }
        public void setEmail(string email)
        {
            this.email = email;
        }
        public string getEmail()
        {
            return email;
        }
        public void showCustomer()
        {
            Console.WriteLine("Id: {0}, Name: {1}, Email: {2}", getId(), getName(),
getEmail());
        }
        public double get_discount()
        {
            return 0;
        }
    }
    class PremiumCustomer : Customer
    {
        private double discount;
        public PremiumCustomer() : base()
        {
        }
        public PremiumCustomer(string name, string email, double discount) : base(name,
email)
        {
            this.discount = discount;
        }
        new public double get_discount()
        {
            return discount;
        }
        new public void showCustomer()
        {
            Console.WriteLine("Id: {0}, Name: {1}, Email: {2}, Premium Discount:{3}%",
getId(), getName(), getEmail(), get_discount());
        }
    }
    class Order
    {
        private static int Number = 0;
        private int id;
        private Customer customer;
        private List<Product> products;
        private double total;
        public Order()
        {
            Number++;
            this.id = Number;
        }
```

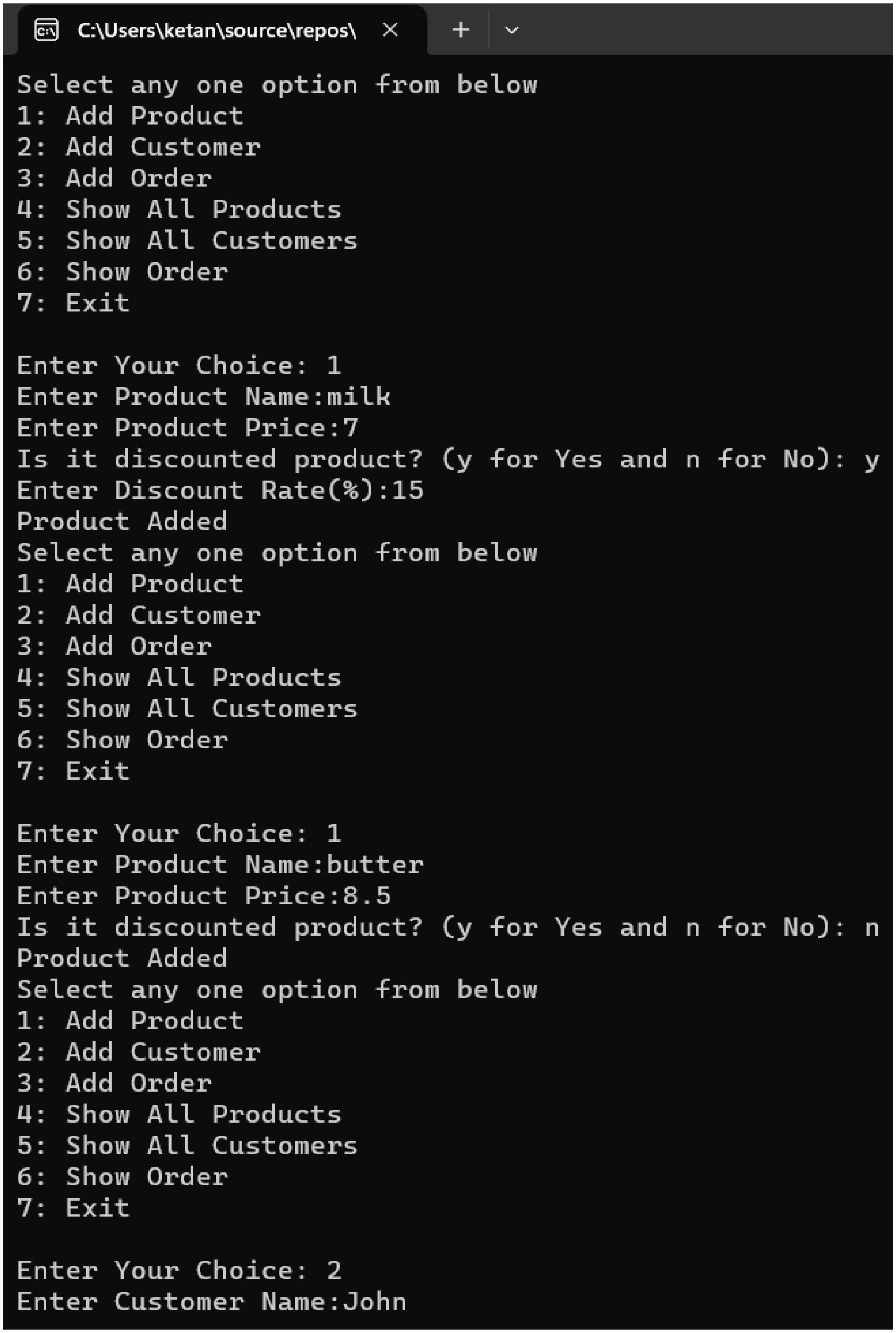
```
public Order(Customer customer, List<Product> products)
{
    Number++;
    this.id = Number;
    this.customer = customer;
    this.products = products;
}
public int getId()
{
    return this.id;
}
public void setCustomer(Customer customer)
{
    this.customer = customer;
}
public Customer getCustomer()
{
    return this.customer;
}
public void setProduct(List<Product> products)
{
    this.products = products;
}
public List<Product> getProduct()
{
    return products;
}
public void calculateTotal()
{
    this.total = 0;
    foreach (Product p in this.products)
    {
        if (p.GetType() == typeof(DiscountedProduct))
        {
            this.total += p.getPrice() - ((DiscountedProduct)p).calculate_discount();
        }
        else
        {
            this.total += p.getPrice() - p.calculate_discount();
        }
    }
    if (customer.GetType() == typeof(PremiumCustomer))
    {
```

```
                this.total = this.total - (this.total *
((PremiumCustomer)customer).get_discount() / 100);
            }
        }
        public double getTotal()
        {
            return total;
        }
        public void showOrder()
        {
            Console.WriteLine("Order Id: {0}", getId());
            Console.WriteLine("==============================");
            Console.WriteLine("Customer's Details");

            if (getCustomer().GetType() == typeof(PremiumCustomer))
            {
                Console.WriteLine("Id: {0}, Name: {1}, Email: {2}, Premium
Discount:{3}%", getCustomer().getId(), getCustomer().getName(),
getCustomer().getEmail(), ((PremiumCustomer)getCustomer()).get_discount());
            }
            else
            {
                Console.WriteLine("Id: {0}, Name: {1}, Email: {2}", getCustomer().getId(),
getCustomer().getName(), getCustomer().getEmail());
            }
            Console.WriteLine("==============================");
            Console.WriteLine("Product's Details");

            Console.WriteLine("{0,-5} {1,-20} {2,-10} {3,-10} {4,-10}", "Id", "Name",
"Price", "Discount", "Net Price");
            foreach (Product p in getProduct())
            {
                if (p.GetType() == typeof(DiscountedProduct))
                {
                    Console.WriteLine("{0,-5} {1,-20} {2,-10} {3,-10} {4,-10}", p.getId(),
p.getName(), p.getPrice(), ((DiscountedProduct)p).calculate_discount(), p.getPrice() -
((DiscountedProduct)p).calculate_discount());
                }
                else
                {
                    Console.WriteLine("{0,-5} {1,-20} {2,-10} {3,-10} {4,-10}", p.getId(),
p.getName(), p.getPrice(), p.calculate_discount(), p.getPrice());
                }
        }
        Console.WriteLine("================================");
        Console.WriteLine("Total Amount: {0}", getTotal());
        Console.WriteLine("================================");
    }
}
class Store
{
    private List<Product> products;
    private List<Customer> customers;
    private List<Order> orders;
    public Store()
    {
        products = new List<Product>();
        customers = new List<Customer>();
        orders = new List<Order>();
    }
    public void addProduct(Product p)
    {
        products.Add(p);
    }
    public void addCustomer(Customer c)
    {
        customers.Add(c);
    }
    public void addOrder(Order o)
    {
        orders.Add(o);
    }
    public void fetchproducts()
    {
        foreach (Product p in products)
        {
            if (p.GetType() == typeof(DiscountedProduct))
            {
                ((DiscountedProduct)p).showProduct();
            }
            else
            {
                p.showProduct();
            }
        }
    }
```

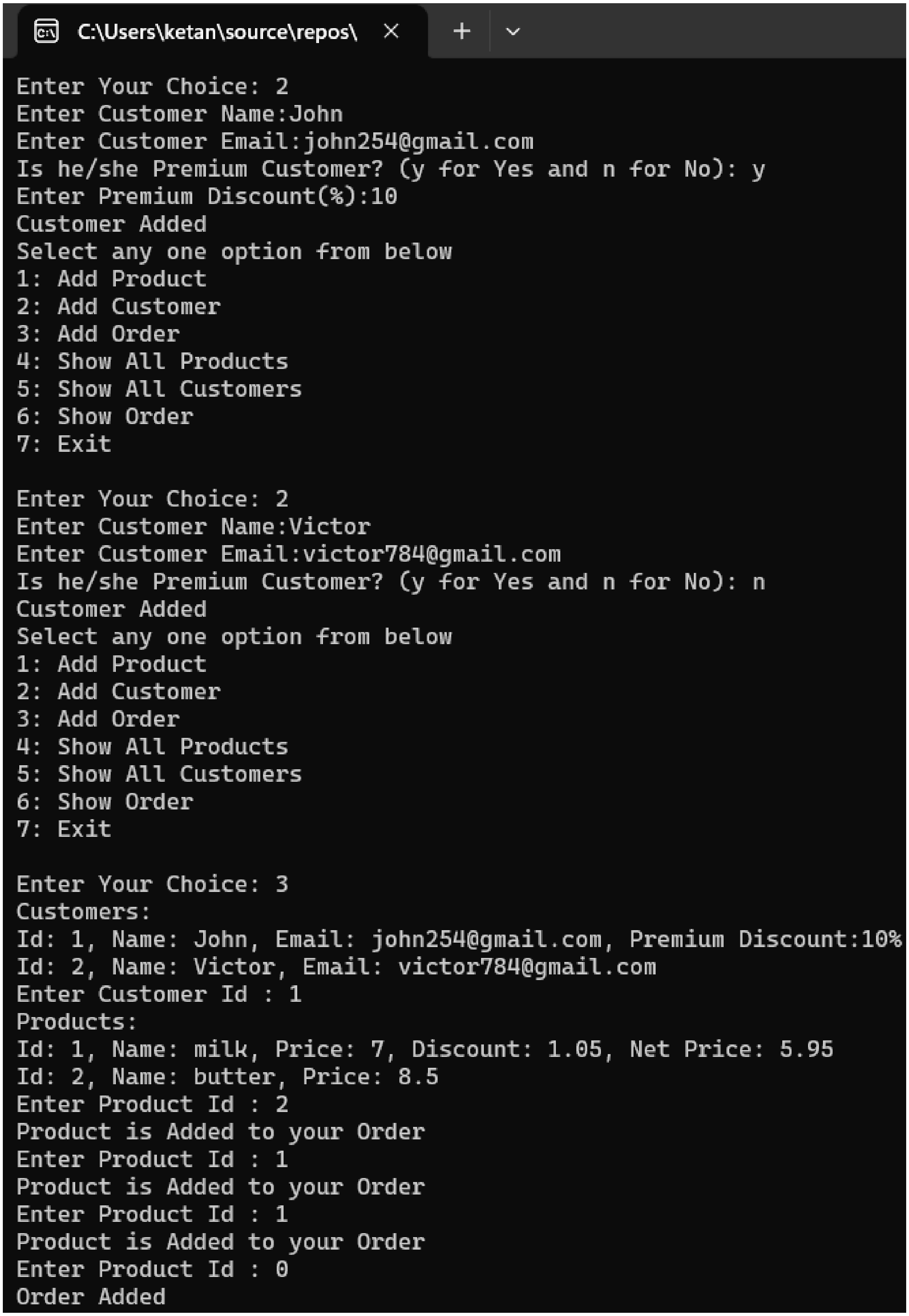
```
public void fetchcustomers()
{
    foreach (Customer c in customers)
    {
        if (c.GetType() == typeof(PremiumCustomer))
        {
            ((PremiumCustomer)c).showCustomer();
        }
        else
        {
            c.showCustomer();
        }
    }
}
public void fetchorders()
{
    Console.WriteLine("List of orders are");
    foreach (Order o in orders)
    {
        o.showOrder();
    }
}
public Product getProductById(int id)
{
    foreach (Product p in products)
    {
        if (p.getId() == id)
        {
            return p;
        }
    }
    return null;
}
public Customer getCustomerById(int id)
{
    foreach (Customer c in customers)
    {
        if (c.getId() == id)
        {
            return c;
        }
    }
    return null;
```

```
        }
        public Order getOrderById(int id)
        {
            foreach (Order o in orders)
            {
                if (o.getId() == id)
                {
                    return o;
                }
            }
            return null;
        }
    }
}
```

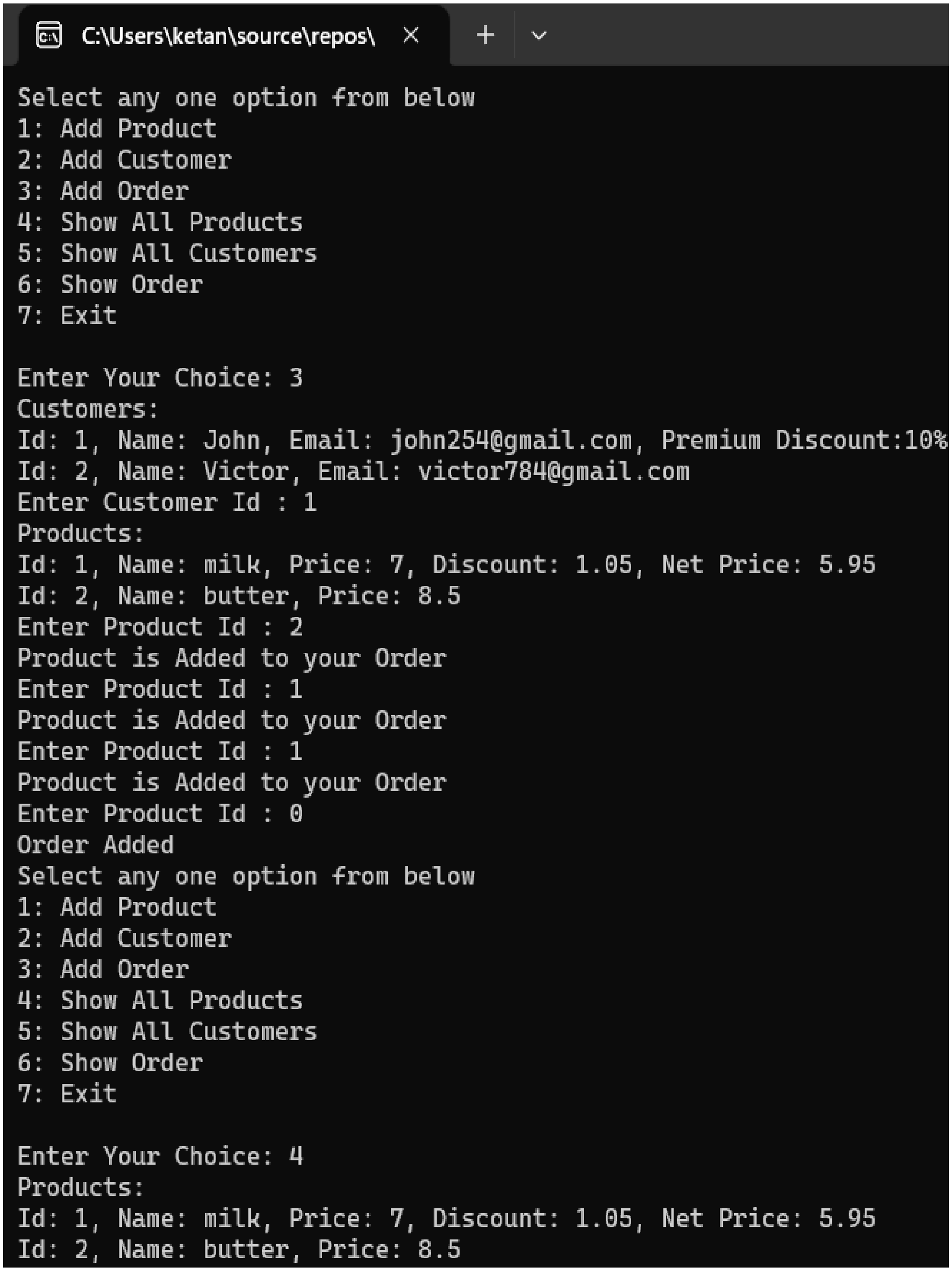
Output:



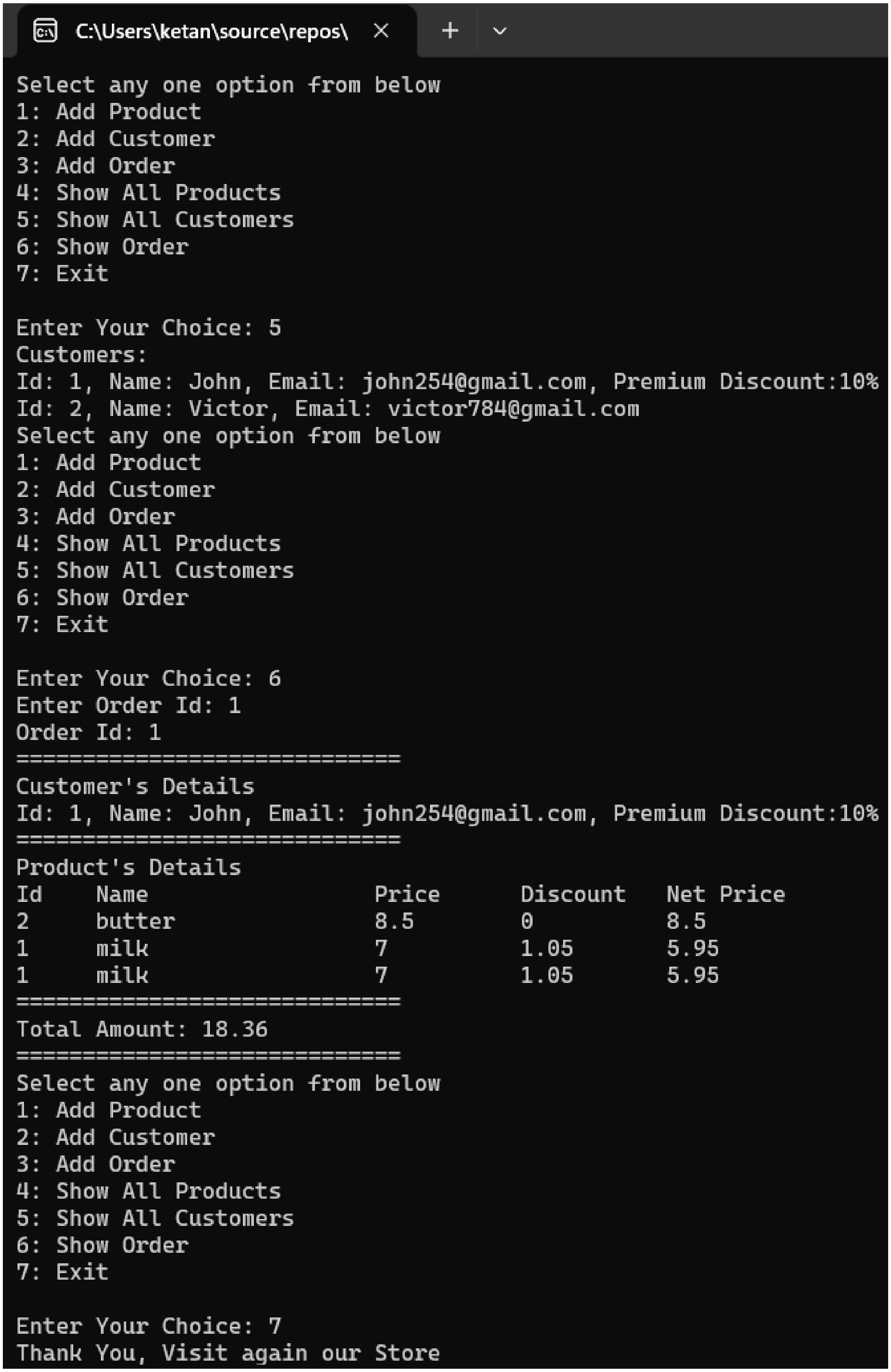
```
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 1
Enter Product Name:milk
Enter Product Price:7
Is it discounted product? (y for Yes and n for No): y
Enter Discount Rate(%):15
Product Added
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 1
Enter Product Name:butter
Enter Product Price:8.5
Is it discounted product? (y for Yes and n for No): n
Product Added
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 2
Enter Customer Name:John
```



```
Enter Your Choice: 2
Enter Customer Name:John
Enter Customer Email:john254@gmail.com
Is he/she Premium Customer? (y for Yes and n for No): y
Enter Premium Discount(%):10
Customer Added
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 2
Enter Customer Name:Victor
Enter Customer Email:victor784@gmail.com
Is he/she Premium Customer? (y for Yes and n for No): n
Customer Added
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 3
Customers:
Id: 1, Name: John, Email: john254@gmail.com, Premium Discount:10%
Id: 2, Name: Victor, Email: victor784@gmail.com
Enter Customer Id : 1
Products:
Id: 1, Name: milk, Price: 7, Discount: 1.05, Net Price: 5.95
Id: 2, Name: butter, Price: 8.5
Enter Product Id : 2
Product is Added to your Order
Enter Product Id : 1
Product is Added to your Order
Enter Product Id : 1
Product is Added to your Order
Enter Product Id : 0
Order Added
```



```
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 3
Customers:
Id: 1, Name: John, Email: john254@gmail.com, Premium Discount:10%
Id: 2, Name: Victor, Email: victor784@gmail.com
Enter Customer Id : 1
Products:
Id: 1, Name: milk, Price: 7, Discount: 1.05, Net Price: 5.95
Id: 2, Name: butter, Price: 8.5
Enter Product Id : 2
Product is Added to your Order
Enter Product Id : 1
Product is Added to your Order
Enter Product Id : 1
Product is Added to your Order
Enter Product Id : 0
Order Added
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 4
Products:
Id: 1, Name: milk, Price: 7, Discount: 1.05, Net Price: 5.95
Id: 2, Name: butter, Price: 8.5
```



```
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 5
Customers:
Id: 1, Name: John, Email: john254@gmail.com, Premium Discount:10%
Id: 2, Name: Victor, Email: victor784@gmail.com
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 6
Enter Order Id: 1
Order Id: 1
=============================
Customer's Details
Id: 1, Name: John, Email: john254@gmail.com, Premium Discount:10%
=============================
Product's Details
Id    Name                 Price      Discount   Net Price
2     butter               8.5        0          8.5
1     milk                 7          1.05       5.95
1     milk                 7          1.05       5.95
=============================
Total Amount: 18.36
=============================
Select any one option from below
1: Add Product
2: Add Customer
3: Add Order
4: Show All Products
5: Show All Customers
6: Show Order
7: Exit

Enter Your Choice: 7
Thank You, Visit again our Store
```