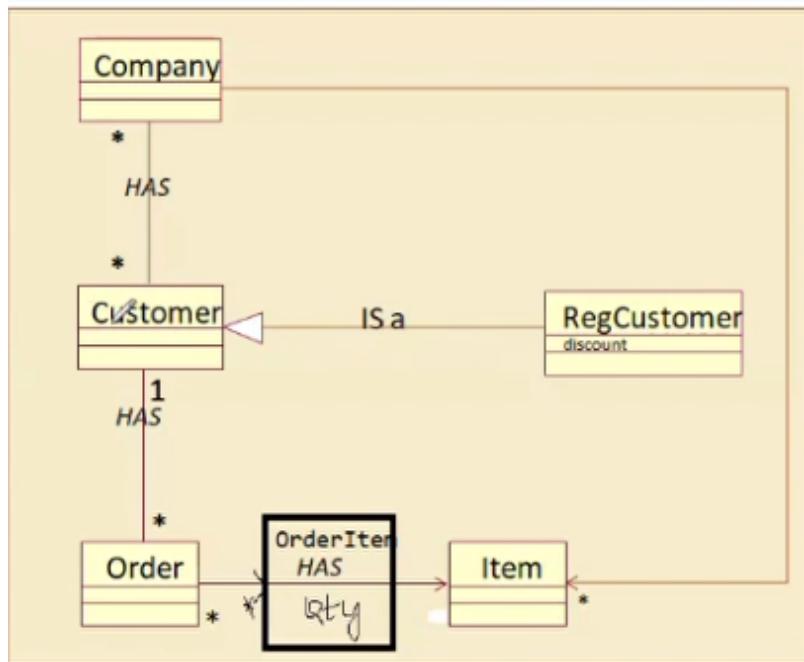


## C#-14 (Company Order)



# Configure your new project

Console App (.NET Framework)    C#    Windows    Console

Project name

CompanyOrder

Location

C:\Shashi\Training\Clients\2021\Mphasis\Oct\_Batch\Demos\OODemos\OODemo



Solution name

CompanyOrder

Place solution and project in the same directory

Framework

.NET Framework 4.5

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search (Ctrl+Q) field. The title bar says "CompanyOrder". The toolbar contains icons for opening, saving, and running the project. The main code editor window displays the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

The Solution Explorer on the right shows the project structure:

- Solution 'CompanyOrder' (1 of 1 projects)
- CompanyOrder
  - Properties
  - References
  - App.config
  - Program.cs

The status bar at the bottom indicates "Creating project 'CompanyOrder'... project creation successful." and shows system details like "Add to Source Control", "Type here to search", and "10:35 AM 23-10-2021".

now we'll create all the classes which are present in the diagram, we'll start with the class which doesn't have any dependency  
1st Item

and we create properties inside it , to create property we type prop tab tab

The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the code editor displays `Item.cs` with the following content:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      class Item
10     {
11         public string ItemName { get; set; }
12         public float Price { get; set; }
13     }
14 }
15
```

The Solution Explorer window on the right shows the project structure:

- Solution 'CompanyOrder' (1 of 1 projects)
- CompanyOrder
  - Properties
  - References
  - App.config
  - Item.cs
  - Program.cs

On the far right, there are quick access links for Properties, Git Changes, and Notifications.

2nd company

```
Company.cs* Item.cs Program.cs
CompanyOrder Company cu!
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      class Company
10     {
11         public string CompanyName { get; set; }
12         private List<Customer> customers = new List<Customer>();
13         //private
14     }
15 }
16
```

company has company name

it has a private list of customer , we have customer and registered customer so do we have to create property for both ---no

now when you are creating a customer, imagine that in your house somebody will come and invite your father for a marriage, when they are inviting, they are inviting your father, so are you not part of that program , you are part of that why, because father represent everyone ,

that's a reason we are going with father, we are not going with children, so you can always put a father there  
so if i create customer, so customer can hold registered customer inside it right that's a reason we are creating list of customer  
here, not the registered customer  
then company will have list of items  
then we have methods to add and get customer

```
private List<Customer> customers = new List<Customer>();  
private List<Item> items = new List<Item>();
```

```
0 references  
public void AddCustomer(Customer customer)  
{  
    this.customers.Add(customer);  
}
```

```
0 references  
public List<Customer> GetCustomers()  
{  
    return this.customers;  
}
```

then add and get item

```
--  
box  
20      0 references  
21      public List<Customer> GetCustomers()  
22      {  
23          return this.customers;  
24      }  
25      0 references  
26      public void AddItem(Item item)  
27      {  
28          this.items.Add(item);  
29      }  
30      0 references  
31      public List<Item> GetItems()  
32      {  
33          return this.items;|  
34      }  
35  }  
36 }
```

then we have customer

here registered customer is a customer, meaning its like father f = new son();

here father is customer and son is registered customer

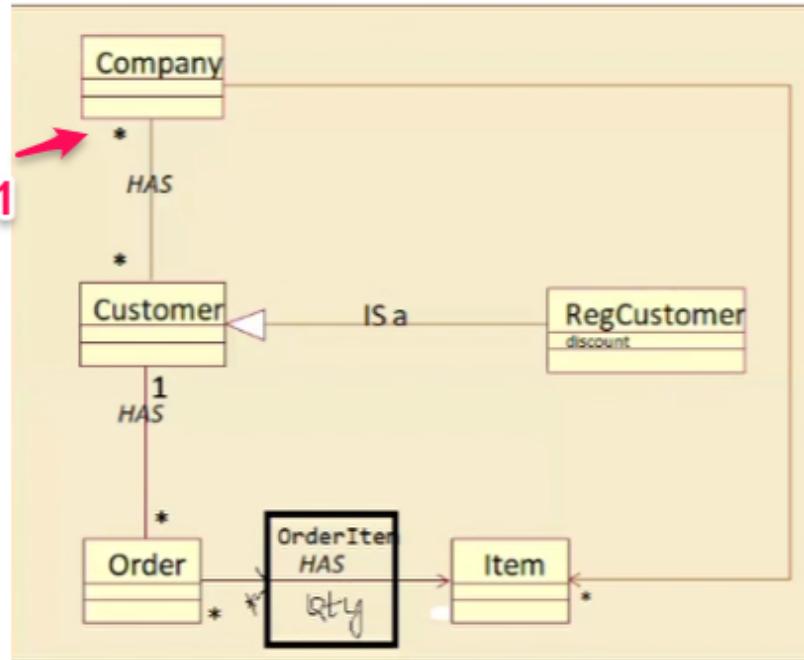
and i can hold the object of son in father

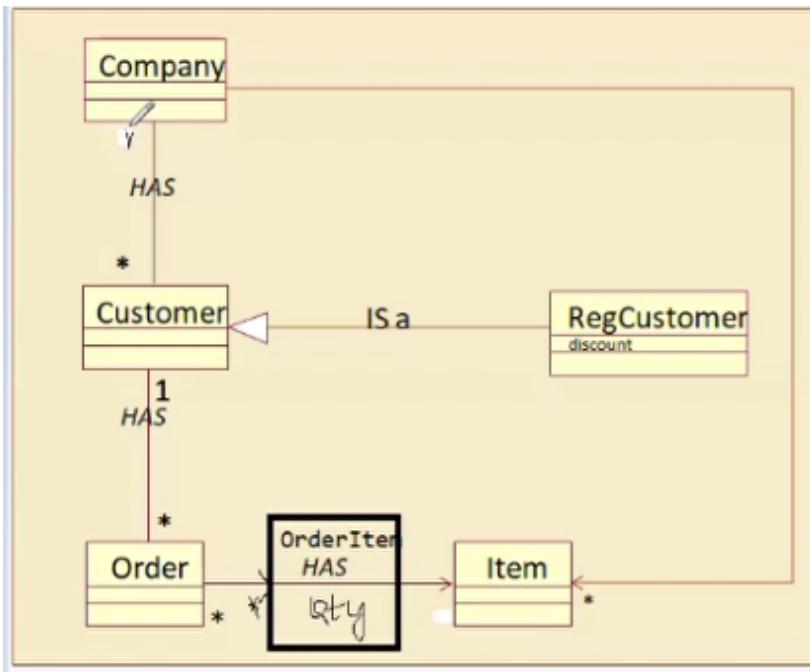
so need to create list here, it doesn't make any sense

now customer will have list of order

and as of now i'm changing the design

instead of  
\* here i'll  
make it as 1





and as we have list of order we need add and get method

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `Company.cs` file, which contains the following C# code:

```
Customer.cs Company.cs Item.cs Program.cs
CompanyOrder
using System.Collections.Generic;
namespace CompanyOrder
{
    class Customer
    {
        public string CustomerName { get; set; }
        public Company Company { get; set; }
        private List<Order> orders = new List<Order>();

        public void AddOrder(Order order)
        {
            this.orders.Add(order);
        }

        public List<Order> GetOrders()
        {
            return this.orders;
        }
}
```

The Solution Explorer on the right side shows the project structure for 'CompanyOrder' with files: `Properties`, `References`, `App.config`, `Company.cs`, `Customer.cs` (which is selected), `Item.cs`, `Order.cs`, and `Program.cs`.

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the `Order.cs` file, which contains the following C# code:

```
using System.Collections.Generic;
namespace CompanyOrder
{
    class Order
    {
        public Customer Customer { get; set; }
        private List<OrderItem> orderItems = new List<OrderItem>();

        public void AddOrderItem(OrderItem orderItem)
        {
            this.orderItems.Add(orderItem);
        }

        public List<OrderItem> GetOrderItems()
        {
            return this.orderItems;
        }
    }
}
```

The Solution Explorer on the right side shows the project structure for `'CompanyOrder'`, which includes files such as `App.config`, `Customer.cs`, `Item.cs`, `Order.cs`, `OrderItem.cs`, and `Program.cs`.

now orderitem

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes 'File', 'Edit', 'View', 'Project', 'Build', 'Tools', 'Help', and 'Recent'. Below the menu is a tab bar with 'OrderItem.cs' (selected), 'Order.cs', 'Customer.cs', 'Company.cs', 'Item.cs', and 'Program.cs'. The main code editor window displays the following C# code:

```
1 namespace CompanyOrder
2 {
3     class OrderItem
4     {
5         public int Qty { get; set; }
6         public Item Item { get; set; }
7     }
8 }
```

The 'Solution Explorer' window on the right lists the project structure:

- CompanyOrder (selected)
- Properties
- References
- App.config
- Company.cs
- Customer.cs
- Item.cs
- Order.cs
- OrderItem.cs (selected)
- Program.cs

now here ek hi item kaise hogा there may be items na  
for multiple items, wo humne order me karliya hai na

 Pay only ₹46,997.00 ₹46,397.00 for this order.  
Apply for Amazon Pay ICICI Bank credit card & get ₹600 back in 60 seconds

[Apply Now](#)

## Shopping Cart

Deselect all items

		Price
<input checked="" type="checkbox"/>	 <b>Mi 11X 5G Cosmic Black 6GB RAM 128GB ROM   SD 870   DisplayMate A+ rated E4 AMOLED   Upto 18 Months No C...</b> In stock Eligible for FREE Shipping <b>a Fulfilled</b> <input type="checkbox"/> This will be a gift <a href="#">Learn more</a> Colour: Cosmic Black Style name: 6GB RAM 128GB Storage Qty: 1 <input type="button" value="Delete"/> <input type="button" value="Save for later"/> <input type="button" value="See more like this"/>	₹27,999.00
<input checked="" type="checkbox"/>	 <b>Samsung Galaxy M12 (Blue,4GB RAM, 64GB Storage) 6000 mAh with 8nm Processor   True 48 MP Quad Camera   90H...</b> <b>#1 Best Seller</b> in Electronics In stock Eligible for FREE Shipping <b>a Fulfilled</b> <input type="checkbox"/> This will be a gift <a href="#">Learn more</a> Colour: Blue Size name: 4GB RAM & 64GB Storage	₹9,499.00

**100% Purchase Protection**  
Original Products | Secure Payments

Your order is eligible for FREE Delivery.  
Select this option at checkout. [Details](#)

**Subtotal (3 items): ₹46,997.00**  
 This order contains a gift

[Proceed to Buy](#)

EMI Available

Explore more items

 **Netgear R6850...**  
★★★★★ 3,646  
₹4,999.00  
[Add to Cart](#)

**Samsung T7 Touch...**

10:47 AM 23-10-2021 

so this is one order and in 1 order you have 2 order items, each order item will have a quantity and item

now registered customer, my reg customer will have a discount

```
RegCustomer.cs*  X  OrderItem.cs  Order.cs  Customer.cs  Company.cs  Item.cs  Program.cs
CompanyOrder
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      0 references
10     class RegCustomer
11     {
12         0 references
13         public int Discout { get; set; }
14     }
15 }
```

Solution Explorer

- Solution 'CompanyOrder' (1 of 1 projects)
- CompanyOrder
  - Properties
  - References
  - App.config
  - Company.cs
  - Customer.cs
  - Item.cs
  - Order.cs
  - OrderItem.cs
  - Program.cs
  - RegCustomer.cs

Properties

Git Changes

Notifications

and it belongs to customer, how will you say it belongs to customer  
how do you write, is a  
: customer , INHERITANCE BABA animal, dog is a animal

The screenshot shows the Microsoft Visual Studio IDE interface. On the left, the Solution Explorer pane displays a solution named 'CompanyOrder' containing a single project also named 'CompanyOrder'. Inside the project, there are several files: Company.cs, Customer.cs, Item.cs, Order.cs, OrderItem.cs, Program.cs, and RegCustomer.cs. The 'RegCustomer.cs' file is currently open in the main code editor window. The code in 'RegCustomer.cs' is as follows:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      class RegCustomer : Customer
10     {
11         public int Discout { get; set; }
12     }
13 }
14
```

The code editor highlights the 'Discout' field with a yellow warning sign icon. The status bar at the bottom of the screen shows the message 'So my class design is ready now we'll write the main'.

So my class design is ready now we'll write the main

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays "CompanyOrder". The menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a search bar. The toolbar contains icons for file operations like Open, Save, and Build. The tabs at the top show RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, and Program.cs (which is selected). The code editor displays the following C# code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace CompanyOrder
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13        }
14    }
15 }
16
```

The status bar at the bottom shows "121 %", "No issues found", "Ln: 1 Ch: 1 TABS CRLF", and a system tray with icons for battery, volume, and network.

RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs

CompanyOrder

0 references

```
11 static void Main(string[] args)
12 {
13     //1. Create Company
14
15     //2. Create Item
16
17     //3. Add item to company
18
19     //4. Create Customer
20
21     //5. Add customer to company
22
23     //6. Add company to customer
24
25     //7. Create order
26
27     //8. Add order to customer
28
29     //9. Add customer to order
30
31     //10. Create orderItem
```

```
--  
33     |           //11. Add orderItem to Order  
34     |           //12. Add item to orderItem  
35     |           }  
36     |       }  
37   }  
38 }  
39
```

```
RegCustomer.cs          OrderItem.cs        Order.cs        Customer.cs      Company.cs       Item.cs        Program.cs*  ✎ X
CompanyOrder.cs          CompanyOrder.cs
CompanyOrder           CompanyOrder.Program
6
7     namespace CompanyOrder
8     {
9         class Program
10    {
11        static void Main(string[] args)
12    {
13        //1. Create Company
14        Company company = new Company() { CompanyName = "Samsung" };
15
16        //2. Create Item
17        Item item1 = new Item() { ItemName = "Note1", Price = 23000 };
18        Item item2 = new Item() { ItemName = "Note2", Price = 22000 };
19        Item item3 = new Item() { ItemName = "Note3", Price = 26000 };
20        Item item4 = new Item() { ItemName = "Note4", Price = 28000 };
21        Item item5 = new Item() { ItemName = "Note5", Price = 25000 };
22        Item item6 = new Item() { ItemName = "Note6", Price = 25000 };
23
24        //3. Add item to company
25        company.AddItem(item1);
26        company.AddItem(item2);
27        company.AddItem(item3);
```

```
//4. Create Customer
Customer customer1 = new Customer() { CustomerName = "Rajesh" };
Customer customer2 = new Customer() { CustomerName = "Ravi" };
Customer customer3 = new RegCustomer() { CustomerName = "Somesh" };
```

we can do this right

father f = new son();

here to this registeredcustomer lets give a fix discount of 10%

```
RegCustomer.cs  X OrderItem.cs  Order.cs  Customer.cs  Company.
CompanyOrder  CompanyOrder.RegC

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      class RegCustomer : Customer
10     {
11         public int Discout { get; set; } = 10;
12     }
13 }
14
```

```
//4. Create Customer
Customer customer1 = new Customer() { CustomerName = "Rajesh" };
Customer customer2 = new Customer() { CustomerName = "Ravi" };
Customer customer3 = new RegCustomer() { CustomerName = "Somesh" };

//5. Add customer to company
company.AddCustomer(customer1);
company.AddCustomer(customer2);
company.AddCustomer(customer3);

//6. Add company to customer
customer1.Company = company;
customer2.Company = company;
customer3.Company = company;
```

```
//7. Create order
Order order1 = new Order();
Order order2 = new Order();
Order order3 = new Order();
Order order4 = new Order();
Order order5 = new Order();
Order order6 = new Order();
```

```
//8. Add order to customer
customer1.AddOrder(order1);
customer1.AddOrder(order2);
customer2.AddOrder(order3);
customer2.AddOrder(order4);
customer3.AddOrder(order5);
```



```
//8. Add order to customer
customer1.AddOrder(order1);
customer1.AddOrder(order2);
customer2.AddOrder(order3);
customer2.AddOrder(order4);
customer3.AddOrder(order5);
customer3.AddOrder(order6);

//9. Add customer to order
order1.Customer = customer1;
order2.Customer = customer1;
order3.Customer = customer2;
order4.Customer = customer2;
order5.Customer = customer3;
order6.Customer = customer3;
```

```
//10. Create orderItem
OrderItem orderItem1 = new OrderItem() { Qty = 1 };
OrderItem orderItem2 = new OrderItem() { Qty = 1 };
OrderItem orderItem3 = new OrderItem() { Qty = 2 };
OrderItem orderItem4 = new OrderItem() { Qty = 2 };
OrderItem orderItem5 = new OrderItem() { Qty = 1 };
OrderItem orderItem6 = new OrderItem() { Qty = 4 };
OrderItem orderItem7 = new OrderItem() { Qty = 3 };
OrderItem orderItem8 = new OrderItem() { Qty = 2 };
OrderItem orderItem9 = new OrderItem() { Qty = 1 };
```

```
93 //11. Add orderItem to Order
94 order1.AddOrderItem(orderItem1);
95 order5.AddOrderItem(orderItem2);
96 order2.AddOrderItem(orderItem3);
97 order6.AddOrderItem(orderItem4);
98 order2.AddOrderItem(orderItem5);
99 order3.AddOrderItem(orderItem6);
100 order3.AddOrderItem(orderItem7);
101 order4.AddOrderItem(orderItem8);
102 order4.AddOrderItem(orderItem9);
103
104
105
106
107
108
//12. Add item to orderItem
orderItem1.Item = item1;
orderItem2.Item = item2;
orderItem3.Item = item1;
orderItem4.Item = item3;
orderItem5.Item = item4;
orderItem6.Item = item5;
orderItem7.Item = item1;
orderItem8.Item = item3;
orderItem9.Item = item6;
orderItem1.Item = item1;
```

The screenshot shows a code editor with multiple tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, Program.cs\*, and a tab for CompanyOrder.Program. The CompanyOrder.cs file is open, displaying the following code:

```
97
98     orderItem4.Item = item5;
99     orderItem5.Item = item4;
100    orderItem6.Item = item5;
101    orderItem7.Item = item1;
102    orderItem8.Item = item3;
103    orderItem9.Item = item6;
104
105    DisplayCompanyDetails(company);
106
107    private static void DisplayCompanyDetails(Company company)
108    {
109        throw new NotImplementedException();
110    }
111
112    }
113
```

A green vertical bar highlights the first few lines of the code, and a yellow vertical bar highlights the entire class definition. A tooltip "1 reference" is visible near the class definition.

lets try to print the company details like this

File Edit Format View Help

## Company Details

---

Company Name : xxx

---

Customer Name:

Order Details

---

ItemName	Price	Qty	Amount
----------	-------	-----	--------

---

xx	xx	xx	xx
----	----	----	----

xx	xx	xx	xx
----	----	----	----

xx	xx	xx	xx
----	----	----	----

xx	xx	xx	xx
----	----	----	----

---

Total Amount:	xx
---------------	----

---

-

```
103     }
104     DisplayCompanyDetails(company);
105 }
106
107     1 reference
108     private static void DisplayCompanyDetails(Company company)
109     {
110         Console.WriteLine("\t\tCompany Details");
111         DrawLine();
112     }
113     1 reference
114     private static void DrawLine()
115     {
116         for (int i = 0; i < 40; i++)
117         {
118             Console.Write("-");
```

RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs\* ✎ X

CompanyOrder

1 reference

```
104     DisplayCompanyDetails(company);
105 }
106
107 private static void DisplayCompanyDetails(Company company)
108 {
109     Console.WriteLine("\t\tCompany Details");
110     DrawLine();
111     Console.WriteLine("Company Name {0}", company.CompanyName);
112     DrawLine();
113     foreach (var customer in company.GetCustomers())
114     {
115         Console.WriteLine("Customer Name " + customer.CustomerName);
116         DrawLine();
117         Console.WriteLine("Item Name\tPrice\tQty\tAmount");
118         DrawLine();
119         foreach (var order in customer.GetOrders())
120         {
121             I
122         }
123     }
124     DrawLine();
```

RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs X

CompanyOrder

CompanyOrder.Program

DisplayCompa

```
115 Console.WriteLine("Customer Name :" + customer.CustomerName);
116 DrawLine();
117 Console.WriteLine("Item Name\tPrice\tQty\tAmount");
118 DrawLine();
119 foreach (var order in customer.GetOrders())
120 {
121     foreach (var orderItem in order.GetOrderItems())
122     {
123         Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
124                         orderItem.Item.Price + "\t" +
125                         orderItem.Qty + "\t" +
126                         (orderItem.Item.Price * orderItem.Qty));
127     }
128 }
129 }
130 DrawLine();
131 }
132
133 private static void DrawLine()...
134
135 }
```

5 references

121 % No issues found

```
Console.WriteLine("Customer Name : " + customerName);
Console.WriteLine("Company Details");
Console.WriteLine("Company Name Samsung");
Console.WriteLine("Customer Name : Rajesh");
Console.WriteLine("-----");
Console.WriteLine("Item Name      Price    Qty     Amount");
Console.WriteLine("-----");
Console.WriteLine("Note1          23000    1       23000");
Console.WriteLine("Note1          23000    2       46000");
Console.WriteLine("Note4          28000    1       28000");
Console.WriteLine("Customer Name : Ravi");
Console.WriteLine("-----");
Console.WriteLine("Item Name      Price    Qty     Amount");
Console.WriteLine("-----");
Console.WriteLine("Note5          25000    4       100000");
Console.WriteLine("Note1          23000    3       69000");
Console.WriteLine("Note3          26000    2       52000");
Console.WriteLine("Note6          25000    1       25000");
Console.WriteLine("Customer Name : Somesh");
Console.WriteLine("-----");
Console.WriteLine("Item Name      Price    Qty     Amount");
Console.WriteLine("-----");
Console.WriteLine("Note2          22000    1       22000");
Console.WriteLine("Note3          26000    2       52000");
Console.WriteLine("-----");
Console.WriteLine("Press any key to continue . . .");

Manager Console  Error List  Immediate Window
```

now i want to get total of amount

i'll create a variable totalAmount and i'll initialize this to 0

RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs X

CompanyOrder

CompanyOrder.Program

DisplayCo

```
115     {
116         Console.WriteLine("Customer Name :" + customer.CustomerName);
117         DrawLine();
118         Console.WriteLine("Item Name\tPrice\tQty\tAmount");
119         DrawLine();
120         foreach (var order in customer.GetOrders())
121         {
122             foreach (var orderItem in order.GetOrderItems())
123             {
124                 Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
125                               orderItem.Item.Price + "\t" +
126                               orderItem.Qty + "\t" +
127                               (orderItem.Item.Price * orderItem.Qty));
128                 totalAmount += orderItem.Item.Price * orderItem.Qty;
129             }
130             DrawLine();
131             Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
132             DrawLine();
133         }
134         DrawLine();
135     }
136     DrawLine();
137 }
138 
```

121 % ✓ No issues found

Output Package Manager Console Error List ... Immediate Window

and at line 128 ill specify this

Company Details			
-----			
Company Name Samsung			
-----			
Customer Name :Rajesh			
-----			
Item Name	Price	Qty	Amount
Note1	23000	1	23000
-----			
Total Amount :			23000
-----			
Note1	23000	2	46000
Note4	28000	1	28000
-----			
Total Amount :			97000
-----			
Customer Name :Ravi			
-----			
Item Name	Price	Qty	Amount
Note5	25000	4	100000
Note1	23000	3	69000
-----			
Total Amount :			266000
-----			
Note3	26000	2	52000
Note6	25000	1	25000
-----			

now here it is taking previous total as well so we need to make total to 0 again  
so see line no 133

A screenshot of the Microsoft Visual Studio IDE interface. The title bar shows "CompanyOrder". The menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search (Ctrl+Q) field. The toolbar contains various icons for file operations like Open, Save, and Build. The solution explorer on the left lists files: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, and Program.cs (the active file). The code editor displays the following C# code:

```
115     {
116         Console.WriteLine("Customer Name :" + customer.CustomerName);
117         DrawLine();
118         Console.WriteLine("Item Name\tPrice\tQty\tAmount");
119         DrawLine();
120         foreach (var order in customer.GetOrders())
121         {
122             foreach (var orderItem in order.GetOrderItems())
123             {
124                 Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
125                     orderItem.Item.Price + "\t" +
126                     orderItem.Qty + "\t" +
127                     (orderItem.Item.Price * orderItem.Qty));
128                 totalAmount += orderItem.Item.Price * orderItem.Qty;
129             }
130             DrawLine();
131             Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
132             DrawLine();
133             totalAmount = 0;
134         }
135         DrawLine();
136     }
137     DrawLine();
138 }
```

The status bar at the bottom shows "121 %", "No issues found", and "Ln:".

```
g- Select C:\WINDOWS\system32\cmd.exe
Total Amount : 23000
-----
Note1 23000 2 46000
Note4 28000 1 28000
-----
Total Amount : 74000
-----
Customer Name :Ravi
-----
Item Name Price Qty Amount
-----
Note5 25000 4 100000
Note1 23000 3 69000
-----
Total Amount : 169000
-----
Note3 26000 2 52000
Note6 25000 1 25000
>-----
>Total Amount : 77000
-----
Customer Name :Somesh
-----
Item Name Price Qty Amount
-----
Note2 22000 1 22000
-----
```

now i need total customer value as well, so rajesh's total value would be 23000 + 74000

Select C:\WINDOWS\system32\cmd.exe

```
Company Details
-----
Company Name Samsung
-----
Customer Name :Rajesh
-----
Item Name      Price    Qty    Amount
-----
Note1          23000   1       23000
-----
Total Amount :           23000
-----
Note1          23000   2       46000
Note4          28000   1       28000
-----
Total Amount :           74000
-----
Total Customer Value :
-----
Customer Name :Ravi
-----
Item Name      Price    Qty    Amount
-----
Note5          25000   4       100000
Note1          23000   3       69000
-----
Total Amount :           169000
-----
Note3          26000   2       52000
```

```
1 reference
7     private static void DisplayCompanyDetails(Company company)
8     {
9         float totalAmount = 0;
0         float customerTotalAmount = 0;
1         Console.WriteLine("\t\tCompany Details");
2         DrawLine();
3         Console.WriteLine("Company Name {0}", company.CompanyName);
4         DrawLine();
5         foreach (var customer in company.GetCustomers())
6         {
7             Console.WriteLine("Customer Name :" + customer.CustomerName);
8             DrawLine();
9             Console.WriteLine("Item Name\tPrice\tQty\tAmount");
0             DrawLine();
1             foreach (var order in customer.GetOrders())
2             {
3                 foreach (var orderItem in order.GetOrderItems())
4                     ;
5             }
6         }
7     }
8 
```

0 1 ← → ⌂

Package Manager Console Error List ... Immediate Window

aded

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays "CompanyOrder". The menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a search bar "Search (Ctrl+Q)". The toolbar contains various icons for file operations like Open, Save, Print, and Find. The solution explorer on the left lists files: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, and Program.cs\*. The "Program.cs\*" tab is selected, showing the following C# code:

```
118     DrawLine();
119     Console.WriteLine("Item Name\tPrice\tQty\tAmount");
120     DrawLine();
121     foreach (var order in customer.GetOrders())
122     {
123         foreach (var orderItem in order.GetOrderItems())
124         {
125             Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                             orderItem.Item.Price + "\t" +
127                             orderItem.Qty + "\t" +
128                             (orderItem.Item.Price * orderItem.Qty));
129             totalAmount += orderItem.Item.Price * orderItem.Qty;
130         }
131         DrawLine();
132         Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
133         DrawLine();
134         customerTotalAmount += totalAmount;
135         totalAmount = 0;
136     }
137     DrawLine();
138     Console.WriteLine("Total Customer Value : \t" + customerTotalAmount );
139     DrawLine();
140 }
```

The status bar at the bottom shows "Ln: 135 Ch: 22 Col: 37 TABS C".

```
CONSOLE.WINTELNET 1 Item Name
Select C:\WINDOWS\system32\cmd.exe

Note1      23000  2      46000
Note4      28000  1      28000
-----
Total Amount : 74000
-----
Total Customer Value : 97000
-----
Customer Name :Ravi
-----
Item Name    Price   Qty     Amount
-----
Note5      25000  4      100000
Note1      23000  3      69000
-----
Total Amount : 169000
-----
Note3      26000  2      52000
Note6      25000  1      25000
-----
Total Amount : 77000
-----
Total Customer Value : 343000
-----
Customer Name :Somesh
-----
Item Name    Price   Qty     Amount
-----
Note2      22000  1      22000
```

but again it is taking the previous value as well

at line 139 we have to make customerTotalAmount = 0

```
  Select C:\WINDOWS\system32\cmd.exe
Note5      25000   4     100000
Note1      23000   3     69000
-----
Total Amount :           169000
-----
Note3      26000   2     52000
Note6      25000   1     25000
-----
Total Amount :           77000
-----
Total Customer Value :  246000
-----
Customer Name :Somesh
-----
Item Name    Price   Qty    Amount
Note2      22000   1     22000
-----
Total Amount :           22000
-----
Note3      26000   2     52000
-----
Total Amount :           52000
-----
Total Customer Value :  74000
-----
Press any key to continue . . .
```

now its fine but somesh should get a discount na as it is a regcustomer  
How do we do that?

The screenshot shows a Microsoft Visual Studio code editor window. The tabs at the top include RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, Program.cs (which is the active tab), and DisplayCor. The code in Program.cs is as follows:

```
124     {
125         Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                           orderItem.Item.Price + "\t" +
127                           orderItem.Qty + "\t" +
128                           (orderItem.Item.Price * orderItem.Qty));
129         totalAmount += orderItem.Item.Price * orderItem.Qty;
130     }
131     DrawLine();
132     Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
133     DrawLine();
134     customerTotalAmount += totalAmount;
135     totalAmount = 0;
136 }
137 DrawLine();
138 Console.WriteLine("Total Customer Value : \t" + customerTotalAmount );
139 customerTotalAmount = 0;
140 DrawLine();
141 }
142 DrawLine();
143 }
144
145 private static void DrawLine()...
```

The line "Console.WriteLine("Total Customer Value : \t" + customerTotalAmount );" is highlighted with a blue selection bar. A mouse cursor is visible over the word "customerTotalAmount". The status bar at the bottom left shows "121 %", a green checkmark icon, and the text "No issues found".

now before printing this totalcustomervalue we need to check if the customer is a registered customer or not

how do you check if the customer object is a registered customer object or not

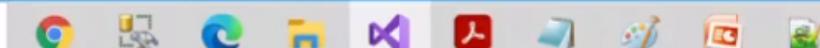
see line 138

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays a C# code editor with the file `Program.cs` open. The code implements a program to calculate the total value of customer orders. It uses nested loops to iterate through customers and their items, calculating the total amount for each customer and the overall total. The code includes methods for drawing lines and handling specific customer types like `RegCustomer`. The code editor features color-coded syntax highlighting and a vertical ruler on the left side. The status bar at the bottom indicates a build succeeded.

```
124     {
125         Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                           orderItem.Item.Price + "\t" +
127                           orderItem.Qty + "\t" +
128                           (orderItem.Item.Price * orderItem.Qty));
129         totalAmount += orderItem.Item.Price * orderItem.Qty;
130     }
131     DrawLine();
132     Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
133     DrawLine();
134     customerTotalAmount += totalAmount;
135     totalAmount = 0;
136 }
137 DrawLine();
138 if(customer is RegCustomer)
139 {
140
141 }
142 Console.WriteLine("Total Customer Value : \t" + customerTotalAmount );
143 customerTotalAmount = 0;
144 DrawLine();
145 }
146 DrawLine();
147 }
```

Build succeeded

Type here to search



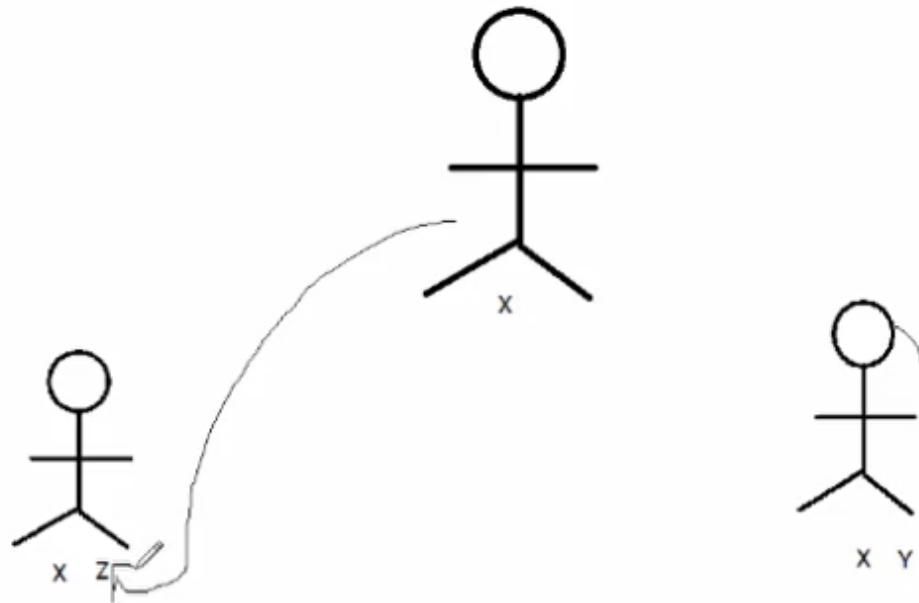
it's like if (animal is dog )  
if it is dog i want to fetch dog's information  
like dog.sound

So here how will i get discount value  
from the customer object i want to get the registered how do you do this

when i say  
father f = new son()  
can i say  
f.z  
from father can i access this z property

```
Son s = new Son();
s.x;
s.z;

Father f = new Son();
f.x;
f.z;
```



not possible

then what i have to do ----- Downcaste it

How to downcaste

```
RegCustomer regCustomer = (RegCustomer)customer;
```

here from customer we had downcaste it to registered customer  
now i'll get discount in registered customer

```
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer. + customer
}
Console.WriteLine("Total Customer");
customerTotalAmount = 0;
DrawLine();

wLine();
| ⌂ Immediate Window
```

so in discount variable i got the discount amount

```
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discout;
}
```

```
RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs X
CompanyOrder.cs CompanyOrder.Program DisplayCompanyDetails(Company

130     }
131     DrawLine();
132     Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
133     DrawLine();
134     customerTotalAmount += totalAmount;
135     totalAmount = 0;
136   }
137   DrawLine();
138   if(customer is RegCustomer)
139   {
140     RegCustomer regCustomer = (RegCustomer)customer;
141     int discount = regCustomer.Discount;
142     float discountTotal = customerTotalAmount - customerTotalAmount * discount / 100;
143     Console.WriteLine("Total Customer Value : \t" + customerTotalAmount);
144   }
145   else
146   {
147     Console.WriteLine("Total Customer Value : \t" + customerTotalAmount);
148   }
149
150   customerTotalAmount = 0;
151   DrawLine();
152 } DrawLine();
```

I'm still getting 74000 for somesh why?

```
  File Edit View Insert Project Tools Help Debug Any CPU
Select C:\WINDOWS\system32\cmd.exe
Total Amount : 169000
-----
Note3      26000  2      52000
Note6      25000  1      25000
-----
Total Amount : 77000
-----
Total Customer Value : 246000
-----
Customer Name : Somesh
-----
Item Name     Price   Qty    Amount
-----
Note2      22000  1      22000
-----
Total Amount : 22000
-----
Note3      26000  2      52000
-----
Total Amount : 52000
-----
Total Customer Value : 74000
-----
Press any key to continue . . .
```

so for reg customer we have to print discount total at line 143

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. The title bar displays "CompanyOrder". The Solution Explorer on the left lists files: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, and Program.cs\*. The Program.cs file is currently selected and open in the main editor area. The code in Program.cs handles the calculation of total customer value, distinguishing between RegCustomer and Company types. The code uses Console.WriteLine for output and DrawLine() for separator lines. A tooltip for the DisplayCompanyDetails method is visible above the code. The status bar at the bottom shows "121 %", "No issues found", "Ln: 143 Ch: 67 Col", and a blue circular icon.

```
130 }  
131 DrawLine();  
132 Console.WriteLine("Total Amount : \t\t\t" + totalAmount);  
133 DrawLine();  
134 customerTotalAmount += totalAmount;  
135 totalAmount = 0;  
136 }  
137 DrawLine();  
138 if(customer is RegCustomer)  
{  
    RegCustomer regCustomer = (RegCustomer)customer;  
    int discount = regCustomer.Discount;  
    float discountTotal = customerTotalAmount - customerTotalAmount * discount / 100;  
    Console.WriteLine("Total Customer Value : \t" + discountTotal);  
}  
144 }  
145 else  
{  
    Console.WriteLine("Total Customer Value : \t" + customerTotalAmount);  
}  
147  
148  
149  
150 customerTotalAmount = 0;  
151 DrawLine();  
152 }  
153 DrawLine();
```

```
8 Select C:\WINDOWS\system32\cmd.exe

Total Amount : 77000
-----
Total Customer Value : 246000
-----
Customer Name : Somesh
-----
Item Name      Price   Qty    Amount
-----
Note2          22000   1      22000
-----
Total Amount : 22000
-----
Note3          26000   2      52000
-----
Total Amount : 52000
-----
Total Customer Value : 66600
-----
Press any key to continue . . .
```

This is Solution 1

now look at some of the problem this program having  
Problem no1: calculating this

```
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                          orderItem.Item.Price + "\t" +
                          orderItem.Qty + "\t" +
                          (orderItem.Item.Price * orderItem.Qty));
        totalAmount += orderItem.Item.Price * orderItem.Qty;
    }
}
```

this is the biggest problem, why because if I change the UI from console application to web application so int the we application also i have to write this line of code  
also this line of code

```
totalAmount += orderItem.Item.Price * orderItem.Qty;
```

all this lines of code i have to write in my ui also now lets fix this  
Now lets try to get this total amount

```
foreach (var orderItem in order.GetOrderItems())
{
    Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                      orderItem.Item.Price + "\t" +
                      orderItem.Qty + "\t" +
                      (orderItem.Item.Price * orderItem.Qty));
    totalAmount += orderItem.Item.Price * orderItem.Qty;
}
```

So who is responsible among all this class  
suppose each class is designed by other person  
someone has designed orderitem.cs

someone orders,

someone customer ..

now whose responsibility is to give me this thing calculated

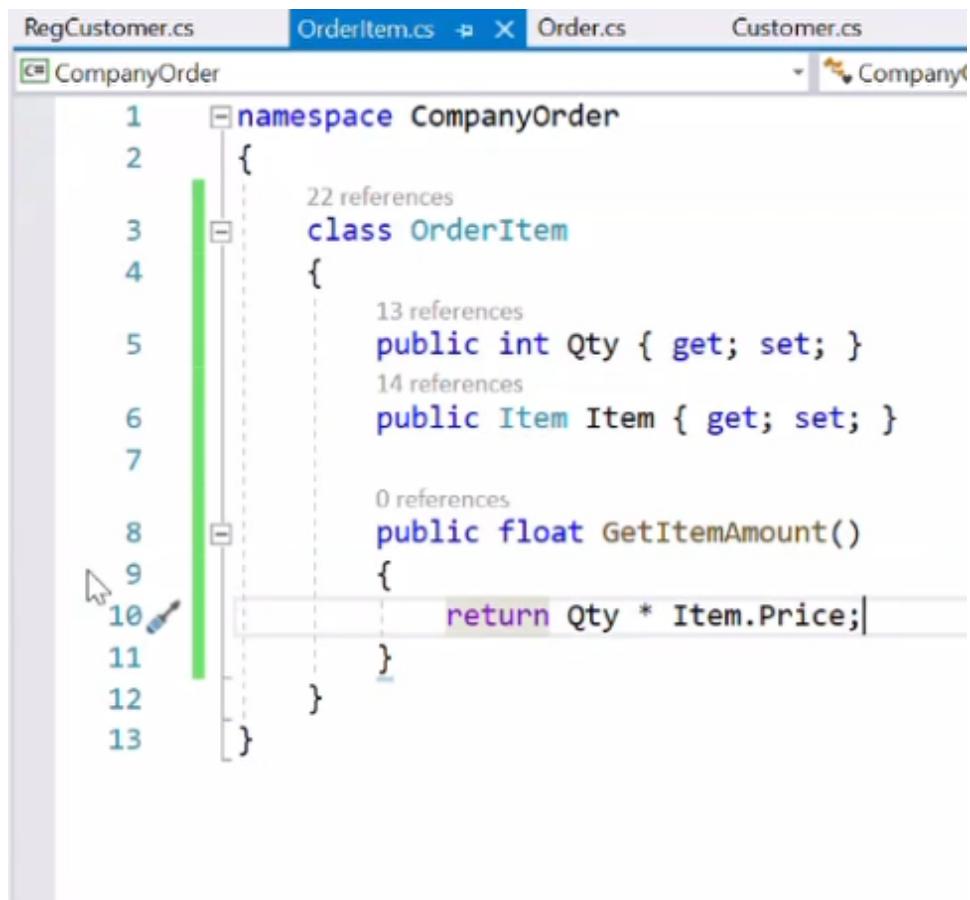
```
(orderItem.Item.Price * orderItem.Qty));
```

```
Item item Price * and
```

which class designer

its a job of orderitem right

so ill go to my orderitem class and introduce a new method here



```
RegCustomer.cs OrderItem.cs Order.cs Customer.cs
CompanyOrder
1  namespace CompanyOrder
2  {
3      class OrderItem
4      {
5          public int Qty { get; set; }
6          public Item Item { get; set; }
7
8          public float GetItemAmount()
9          {
10             return Qty * Item.Price;
11         }
12     }
13 }
```

now my orderitem has the method to calculate orderitem

```
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                          orderItem.Item.Price + "\t" +
                          orderItem.Qty + "\t" +
                          orderItem.GetItemAmount());
        totalAmount += orderItem.Item.Price * orderItem.Qty;
    }
}
```

`totalAmount += orderItem.Item.Price * orderItem.Qty;`

Writes the specified string value, followed by the current line terminator, to the standard output stream. The value is written unless `value` is `null`.

look at this I'm getting the total amount

```
Select C:\WINDOWS\system32\cmd.exe
      Company Details
-----
Company Name Samsung
-----
Customer Name :Rajesh
-----
Item Name      Price   Qty     Amount
-----
Note1          23000   1       23000
-----
Total Amount :                         23000
-----
Note1          23000   2       46000
Note4          28000   1       28000
-----
Total Amount :                         74000
-----
Total Customer Value :  97000
-----
Customer Name :Ravi
-----
Item Name      Price   Qty     Amount
-----
Note5          25000   4       100000
Note1          23000   3       69000
-----
Total Amount :                         169000
-----
Note3          26000   2       52000
```

Now how do we eliminate this total amount  
tell me this totalamount belongs to orderitem, order or customer

```
foreach (var customer in company.GetCustomers())
{
    Console.WriteLine("Customer Name : " + customer.CustomerName);
    DrawLine();
    Console.WriteLine("Item Name\tPrice\tQty\tAmount");
    DrawLine();
    foreach (var order in customer.GetOrders())
    {
        foreach (var orderItem in order.GetOrderItems())
        {
            Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                orderItem.Item.Price + "\t" +
                orderItem.Qty + "\t" +
                orderItem.GetItemAmount());
            totalAmount += orderItem.Item.Price * orderItem.Qty;
        }
        DrawLine();
        Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
        DrawLine();
        customerTotalAmount += totalAmount;
        totalAmount = 0;
    }
}
```

look at the loop yaar

this loop is for orderitem

```
foreach (var customer in company.GetCustomers())
{
    Console.WriteLine("Customer Name :" + customer.CustomerName);
    DrawLine();
    Console.WriteLine("Item Name\tPrice\tQty\tAmount");
    DrawLine();
    foreach (var order in customer.GetOrders())
    {
        foreach (var orderItem in order.GetOrderItems())
        {
            Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                orderItem.Item.Price + "\t" +
                orderItem.Qty + "\t" +
                orderItem.GetItemAmount());
            totalAmount += orderItem.Item.Price * orderItem.Qty;
        }
        DrawLine();
        Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
        DrawLine();
        customerTotalAmount += totalAmount;
        totalAmount = 0;
    }
}
```

this loop is for what, its for order right

```
foreach (var customer in company.GetCustomers())
{
    Console.WriteLine("Customer Name :" + customer.CustomerName);
    DrawLine();
    Console.WriteLine("Item Name\tPrice\tQty\tAmount");
    DrawLine();
    foreach (var order in customer.GetOrders())
    {
        foreach (var orderItem in order.GetOrderItems())
        {
            Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                orderItem.Item.Price + "\t" +
                orderItem.Qty + "\t" +
                orderItem.GetItemAmount());
            totalAmount += orderItem.Item.Price * orderItem.Qty;
        }
        DrawLine();
        Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
        DrawLine();
        customerTotalAmount += totalAmount;
        totalAmount = 0;
    }
}
```

so whose job is to fix this total amount is its order's job or orderitem's job

where are we printing this total amount

this total amount belongs to order right, its the total of the order

```
[Select C:\WINDOWS\system32\cmd.exe]
Total Customer Value : 97000
-----
Customer Name :Ravi
-----
Item Name      Price   Qty    Amount
-----
Note5          25000   4      100000
Note1          23000   3      69000
-----
Total Amount :                      169000
-----
Note3          26000   2      52000
Note6          25000   1      25000
-----
Total Amount :                      77000
-----
Total Customer Value : 246000
-----
Customer Name :Somesh
-----
Item Name      Price   Qty    Amount
-----
Note2          22000   1      22000
-----
Total Amount :                      22000
-----
Note3          26000   2      52000
-----
Total Amount :                      52000
```

I'm adding 169000 + 77000 that's when i get 246000 right  
i'm taking about this total amount

G c# hashmap example - Google S × a Amazon.in Shopping Cart × +

amazon.in/gp/cart/view.html?ref\_=nav\_cart

 Mi 11X 5G Cosmic Black 6GB RAM 128GB ROM | SD 870 | DisplayMate A+ rated E4 AMOLED | Upto 18 Months No C... ₹27,999.00

In stock  
Eligible for FREE Shipping  
**a Fulfilled**  
 This will be a gift [Learn more](#)

Colour: Cosmic Black  
Style name: 6GB RAM 128GB Storage

Qty: 1  Delete Save for later See more like this

---

 Samsung Galaxy M12 (Blue,4GB RAM, 64GB Storage) 6000 mAh with 8nm Processor | True 48 MP Quad Camera | 90H... ₹9,499.00

#1 Best Seller in Electronics  
In stock  
Eligible for FREE Shipping  
**a Fulfilled**  
 This will be a gift [Learn more](#)

Colour: Blue  
Size name: 4GB RAM & 64GB Storage

Qty: 2  Delete Save for later See more like this

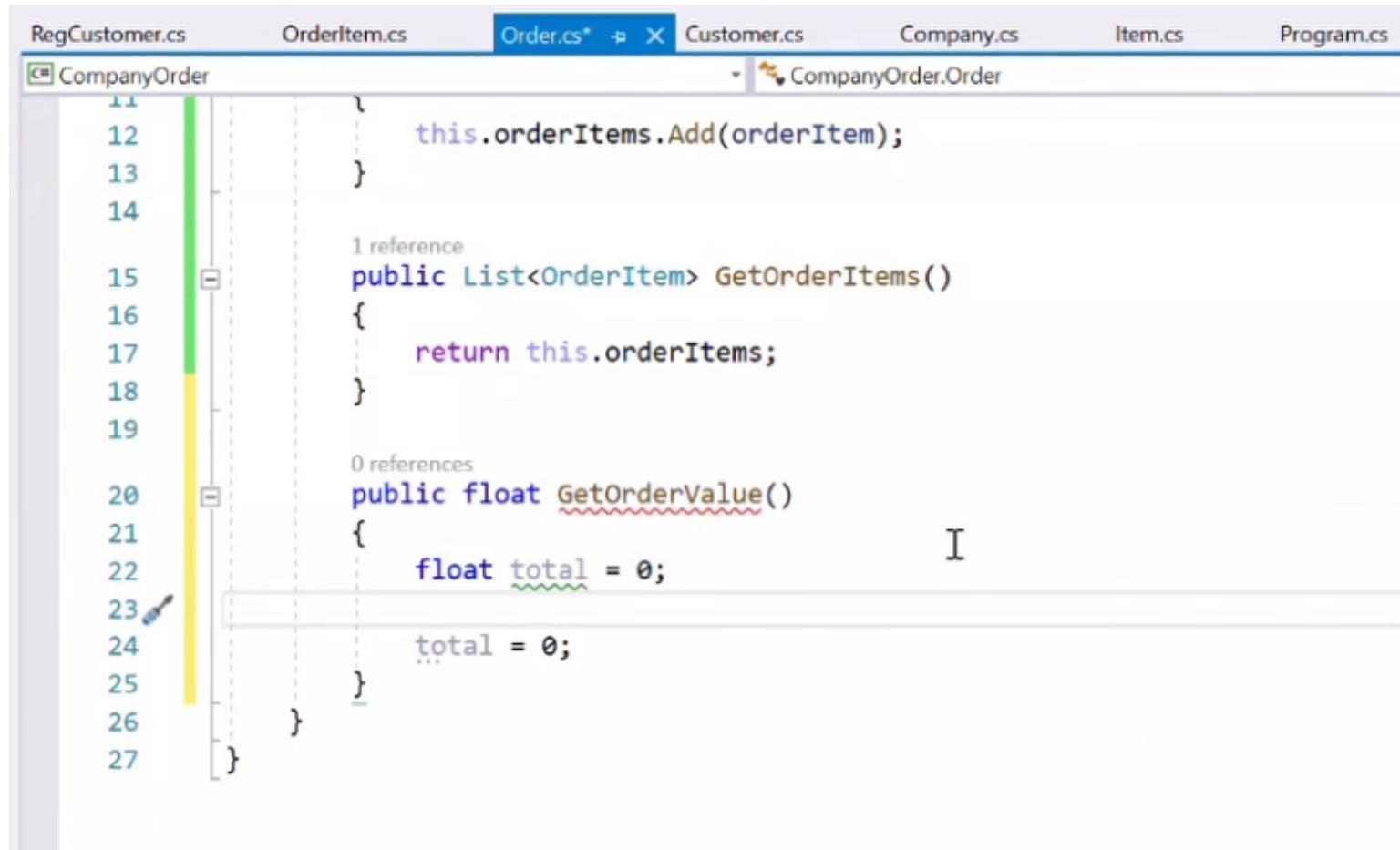
---

Subtotal (3 items): ₹46,997.00

Saved for later (4 items)

So this belongs to what -----order  
so it has to be fix by order guy whoever creates that order class, it is his responsibility to give me this total order value

So i'll go to my order class and i'll introduce a new class GetOrderValue()



```
RegCustomer.cs OrderItem.cs Order.cs* Customer.cs Company.cs Item.cs Program.cs
CompanyOrder
11
12     this.orderItems.Add(orderItem);
13 }
14
15     1 reference
16     public List<OrderItem> GetOrderItems()
17     {
18         return this.orderItems;
19     }
20     0 references
21     public float GetOrderValue()
22     {
23         float total = 0;
24         total = 0;
25     }
26 }
27 }
```

Now how to get the total order value printed

```
    foreach (var customer in company.GetCustomers())
    {
        Console.WriteLine("Customer Name :" + customer.CustomerName);
        DrawLine();
        Console.WriteLine("Item Name\tPrice\tQty\tAmount");
        DrawLine();
        foreach (var order in customer.GetOrders())
        {
            foreach (var orderItem in order.GetOrderItems())
            {
                Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                    orderItem.Item.Price + "\t" +
                    orderItem.Qty + "\t" +
                    orderItem.GetItemAmount());
                totalAmount += orderItem.Item.Price * orderItem.Qty;
            }
            DrawLine();
            Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
            DrawLine();
            customerTotalAmount += totalAmount;
            totalAmount = 0;
        }
        DrawLine();
        if(customer is RepCustomer)
    }
```

Error List Immediate Window

we are adding up this totalamount

## CompanyOrder.Program

```
DrawLine();
Console.WriteLine("Item Name\tPrice\tQty\tAmount");
DrawLine();
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                          orderItem.Item.Price + "\t" +
                          orderItem.Qty + "\t" +
                          orderItem.GetItemAmount());
        totalAmount += orderItem.Item.Price * orderItem.Qty;
    }
    DrawLine();
    Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
    DrawLine();
    customerTotalAmount += totalAmount;
    totalAmount = 0;
}
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discount;
```

so this GetItemAmount what we have here

```
RegCustomer.cs OrderItem.cs Order.cs* Customer.cs Company.cs Item.cs Program.cs
CompanyOrder CompanyOrder.Program D

118     DrawLine();
119     Console.WriteLine("Item Name\tPrice\tQty\tAmount");
120     DrawLine();
121     foreach (var order in customer.GetOrders())
122     {
123         foreach (var orderItem in order.GetOrderItems())
124         {
125             Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                             orderItem.Item.Price + "\t" +
127                             orderItem.Qty + "\t" +
128                             orderItem.GetItemAmount());
129             totalAmount += orderItem.Item.Price * orderItem.Qty;
130         }
131         DrawLine();
132         Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
133         DrawLine();
134         customerTotalAmount += totalAmount;
135         totalAmount = 0;
136     }
137     DrawLine();
138     if(customer is RegCustomer)
139     {
140         RegCustomer regCustomer = (RegCustomer)customer;
141         int discount = regCustomer.Discount;
```

121 % No issues found

we are adding up that one so it is just like this

The screenshot shows a Windows application window titled "CompanyOrder.Program". The main area contains C# code for printing customer orders. The code uses `Console.WriteLine` to output tab-separated values for item name, price, quantity, and amount. It loops through customers and their order items, calculating a total amount. A conditional block checks if the customer is a registered customer and initializes a variable for discount calculation.

```
118     DrawLine();
119     Console.WriteLine("Item Name\tPrice\tQty\tAmount");
120     DrawLine();
121     foreach (var order in customer.GetOrders())
122     {
123         foreach (var orderItem in order.GetOrderItems())
124         {
125             Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                             orderItem.Item.Price + "\t" +
127                             orderItem.Qty + "\t" +
128                             orderItem.GetItemAmount());
129             totalAmount += orderItem.GetItemAmount();
130         }
131         DrawLine();
132         Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
133         DrawLine();
134         customerTotalAmount += totalAmount;
135         totalAmount = 0;
136     }
137     DrawLine();
138     if(customer is RegCustomer)
139     {
140         RegCustomer regCustomer = (RegCustomer)customer;
141         int discount = regCustomer.Discount;
```

meaning my order has many orderitem

i need to loop through all those orderitem and call these method this method will give me amount calculated

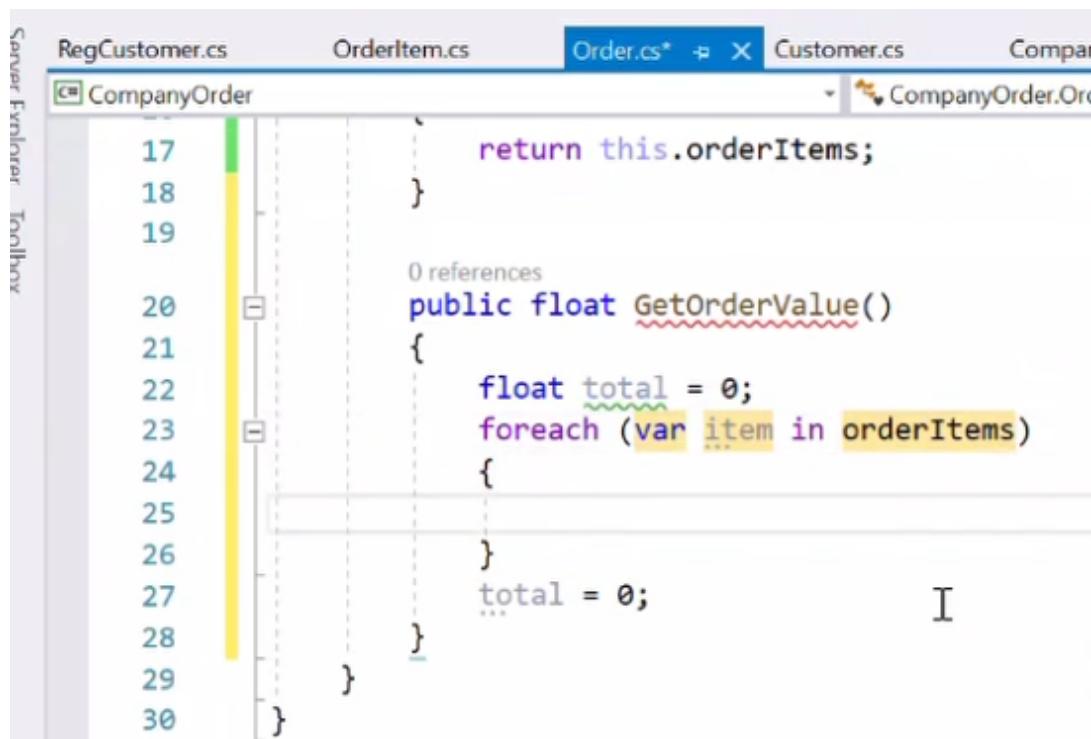
```
DrawLine();
Console.WriteLine("Item Name\tPrice\tQty\tAmount");
DrawLine();
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                          orderItem.Item.Price + "\t" +
                          orderItem.Qty + "\t" +
                          orderItem.GetItemAmount());
        totalAmount += orderItem.GetItemAmount();
    }
    DrawLine();
    Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
    DrawLine();
    customerTotalAmount += totalAmount;
    totalAmount = 0;
}
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discount:
```

so that's what we do, we'll loop through all the orderitems

Orderitems is a list isn't it

```
RegCustomer.cs OrderItem.cs Order.cs* ✘ Customer.cs Company.cs Item.cs Program.cs
CompanyOrder
2
3  namespace CompanyOrder
4  {
5      class Order
6      {
7          public Customer Customer { get; set; }
8          private List<OrderItem> orderItems = new List<OrderItem>();
9
10         public void AddOrderItem(OrderItem orderItem) {
11             this.orderItems.Add(orderItem);
12         }
13
14
15         public List<OrderItem> GetOrderItems()
16         {
17             return this.orderItems;
18         }
19
20         public float GetOrderValue()
21     }
```

so we'll use a foreach where i'll loop through all the order items



```
17     return this.orderItems;
18 }
19
20 public float GetOrderValue()
21 {
22     float total = 0;
23     foreach (var item in orderItems)
24     {
25         ...
26     }
27     total = 0;
28 }
29 }
30 }
```

then i get the item and item has a GetItemAmount which will get me the item amount which we'll return with the total

```
RegCustomer.cs OrderItem.cs Order.cs* < X Customer.cs Company.cs
C# CompanyOrder
17     return this.orderItems;
18 }
19
20 public float GetOrderValue()
21 {
22     float total = 0;
23     foreach (var item in orderItems)
24     {
25         total += item.GetItemAmount();
26     }
27     return total;
28 }
29
30 }
```

So i'll come back here and remove this piece of code, because this is already been calculated

OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs X

CompanyOrder.Program

```
DrawLine();
Console.WriteLine("Item Name\tPrice\tQty\tAmount");
DrawLine();
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                          orderItem.Item.Price + "\t" +
                          orderItem.Qty + "\t" +
                          orderItem.GetItemAmount());
        totalAmount += orderItem.GetItemAmount();
    }
    DrawLine();
    Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
    DrawLine();
    customerTotalAmount += totalAmount;
    totalAmount = 0;
}
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discount:
}
```

and instead of me saying total amount

```
Order.cs      Customer.cs      Company.cs      Item.cs      Program.cs*  ✘ X
              ▾ CompanyOrder.Program
DrawLine();
Console.WriteLine("Item Name\tPrice\tQty\tAmount");
DrawLine();
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                           orderItem.Item.Price + "\t" +
                           orderItem.Qty + "\t" +
                           orderItem.GetItemAmount());
        //totalAmount += orderItem.GetItemAmount();
    }
    DrawLine();
    Console.WriteLine("Total Amount : \t\t\t" + totalAmount);
    DrawLine();
    customerTotalAmount += totalAmount;
    totalAmount = 0;
}
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discount:
```

i'll say order.GetOrderValue()

```
Order.cs      Customer.cs     Company.cs    Item.cs    Program.cs  X
              ▾ CompanyOrder.Program
              ▾ DisplayCo

DrawLine();
Console.WriteLine("Item Name\tPrice\tQty\tAmount");
DrawLine();
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                           orderItem.Item.Price + "\t" +
                           orderItem.Qty + "\t" +
                           orderItem.GetItemAmount());
        //totalAmount += orderItem.GetItemAmount();
    }
    DrawLine();
    Console.WriteLine("Total Amount : \t\t\t" + order.GetOrderValue());
    DrawLine();
    customerTotalAmount += totalAmount;
    totalAmount = 0;
}
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discount;
}

List ... Immediate Window
```

look at this i'm getting the totalordervalue

```
g Select C:\WINDOWS\system32\cmd.exe
g                                         Company Details
Company Name Samsung
Customer Name :Rajesh
Item Name      Price   Qty    Amount
Note1          23000   1      23000
Total Amount :                         23000
Note1          23000   2      46000
Note4          28000   1      28000
Total Amount :                         74000
Total Customer Value :  0
Customer Name :Ravi
Item Name      Price   Qty    Amount
Note5          25000   4      100000
Note1          23000   3      69000
Total Amount :                         169000
Note2          26000   2      52000
```

now how do you get this totalAmount calculated then

RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs X

CompanyOrder CompanyOrder.Program DisplayCompanyDetail

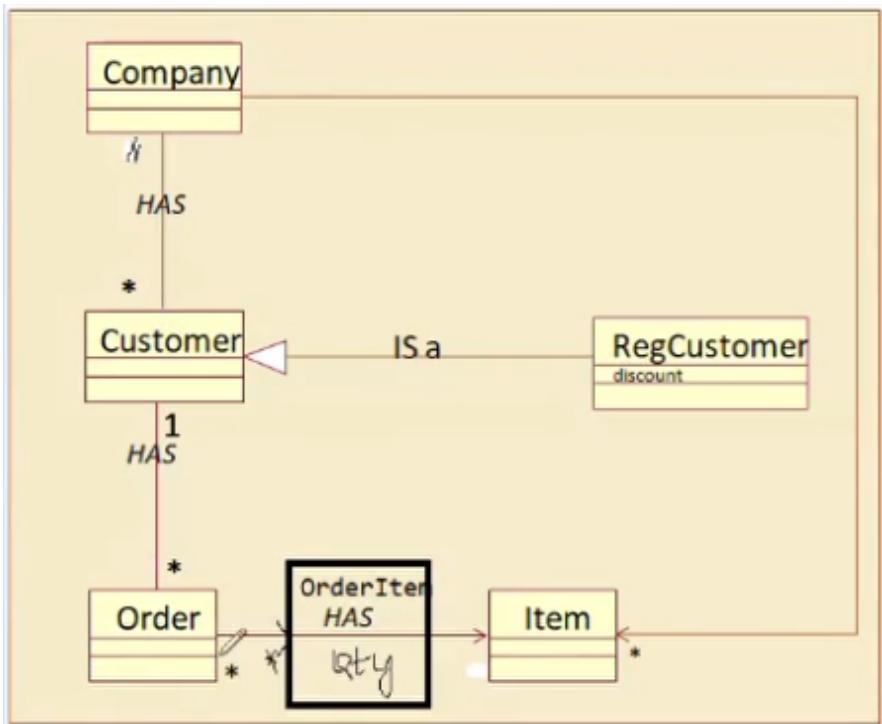
```
118     DrawLine();
119     Console.WriteLine("Item Name\tPrice\tQty\tAmount");
120     DrawLine();
121     foreach (var order in customer.GetOrders())
122     {
123         foreach (var orderItem in order.GetOrderItems())
124         {
125             Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                             orderItem.Item.Price + "\t" +
127                             orderItem.Qty + "\t" +
128                             orderItem.GetItemAmount());
129             //totalAmount += orderItem.GetItemAmount();
130         }
131         DrawLine();
132         Console.WriteLine("Total Amount : \t\t\t" + order.GetOrderValue());
133         DrawLine();
134         customerTotalAmount += totalAmount;
135         totalAmount = 0;
136     }
137     DrawLine();
138     if(customer is RegCustomer)
139     {
140         RegCustomer regCustomer = (RegCustomer)customer;
141         int discount = regCustomer.Discount;
```

121 % ✓ No issues found

Output Package Manager Console Error List Immediate Window

```
  Select C:\WINDOWS\system32\cmd.exe
      Company Details
-----
Company Name Samsung
-----
Customer Name :Rajesh
-----
Item Name      Price   Qty    Amount
-----
Note1          23000   1      23000
-----
Total Amount :                      23000
-----
Note1          23000   2      46000
Note4          28000   1      28000
-----
Total Amount :                      74000
-----
Total Customer Value :  0
-----
Customer Name :Ravi
-----
Item Name      Price   Qty    Amount
-----
Note5          25000   4      100000
Note1          23000   3      69000
-----
Total Amount :                      169000
-----
Note3          26000   2      52000
```

Now whose job is to get the total customer value -----its the guy who wrote customer.cs  
we are in customer class and i want to get the total order value



so this part we have understood

```
CompanyOrder.Program
```

```
Display
```

```
DrawLine();
Console.WriteLine("Item Name\tPrice\tQty\tAmount");
DrawLine();
foreach (var order in customer.GetOrders())
{
    foreach (var orderItem in order.GetOrderItems())
    {
        Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
                          orderItem.Item.Price + "\t" +
                          orderItem.Qty + "\t" +
                          orderItem.GetItemAmount());
        //totalAmount += orderItem.GetItemAmount();
    }
    DrawLine();
    Console.WriteLine("Total Amount : \t\t\t" + order.GetOrderValue())
    DrawLine();
    customerTotalAmount += totalAmount;
    totalAmount = 0;
}
DrawLine();
if(customer is RegCustomer)
{
    RegCustomer regCustomer = (RegCustomer)customer;
    int discount = regCustomer.Discount:
}
```

orderitem.GetItemAmount()

so now each and every individual order we are calculating this amount



Pay only ₹46,997.00 ₹46,397.00 for this order.  
Apply for Amazon Pay ICICI Bank credit card & get ₹600 back in 60 seconds

## Shopping Cart

[Deselect all items](#)

Price



Mi 11X 5G Cosmic Black 6GB RAM 128GB ROM | SD 870 |  
DisplayMate A+ rated E4 AMOLED | Upto 18 Months No C...

₹27,999.00

In stock

Eligible for FREE Shipping

Fulfilled

This will be a gift [Learn more](#)

Colour: Cosmic Black

Style name: 6GB RAM 128GB Storage

Qty: 1

[Delete](#)

[Save for later](#)

[See more like this](#)



Samsung Galaxy M12 (Blue, 4GB RAM, 64GB Storage) 6000 mAh with 8nm Processor | True 48 MP Quad Camera | 90H...

₹9,499.00

#1 Best Seller in Electronics

In stock

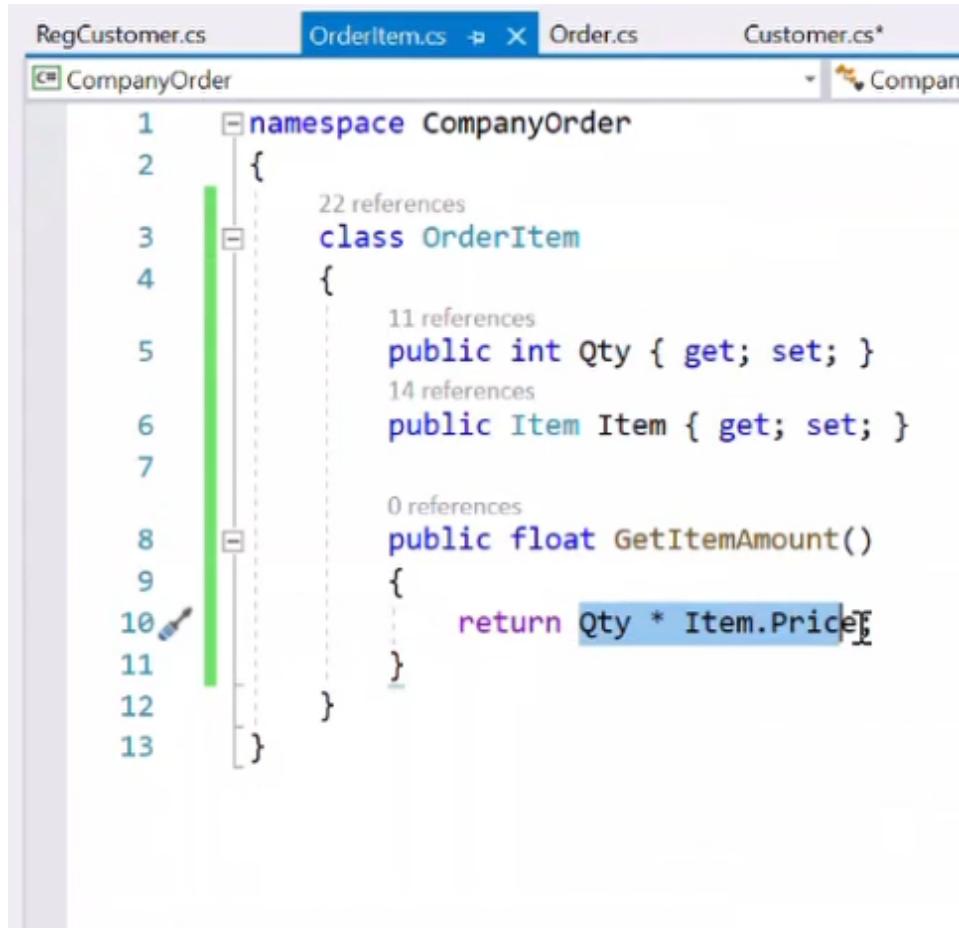


Type here to search



this is a row this is my GetItemAmount()

when i say this, so in my orderitem if you see here  
in the orderitem you will get the quantity \* price



```
RegCustomer.cs OrderItem.cs Order.cs Customer.cs*
CompanyOrder
1  namespace CompanyOrder
2  {
3      class OrderItem
4      {
5          public int Qty { get; set; }
6          public Item Item { get; set; }
7
8          public float GetItemAmount()
9          {
10             return Qty * Item.Price;
11         }
12     }
13 }
```

Now how will you get the total order value

## Shopping Cart

[Deselect all items](#)

Price



Mi 11X 5G Cosmic Black 6GB RAM 128GB ROM | SD 870 |  
DisplayMate A+ rated E4 AMOLED | Upto 18 Months No C...

₹27,999.00

In stock

Eligible for FREE Shipping

Fulfilled

This will be a gift [Learn more](#)

Colour: Cosmic Black

Style name: 6GB RAM 128GB Storage

Qty: 1

[Delete](#)

[Save for later](#)

[See more like this](#)



Samsung Galaxy M12 (Blue, 4GB RAM, 64GB Storage) 6000  
mAh with 8nm Processor | True 48 MP Quad Camera | 90H...

₹9,499.00

#1 Best Seller in Electronics

In stock

Eligible for FREE Shipping

Fulfilled

This will be a gift [Learn more](#)

Colour: Blue

Size name: 4GB RAM & 64GB Storage

Qty: 2

[Delete](#)

[Save for later](#)

[See more like this](#)

Subtotal (3 items): ₹46,997.00

totalordervalue = all this ordervalues what we have here orderitem 1 + orderitem2,  
so i want to add these 2 amounts

So i goto my order , order has many orderitem, one order has many order item

The screenshot shows a code editor with multiple tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs (selected), Customer.cs\*, Company.cs, Item.cs, and Program.cs. The main pane displays the Order.cs file, which contains the following C# code:

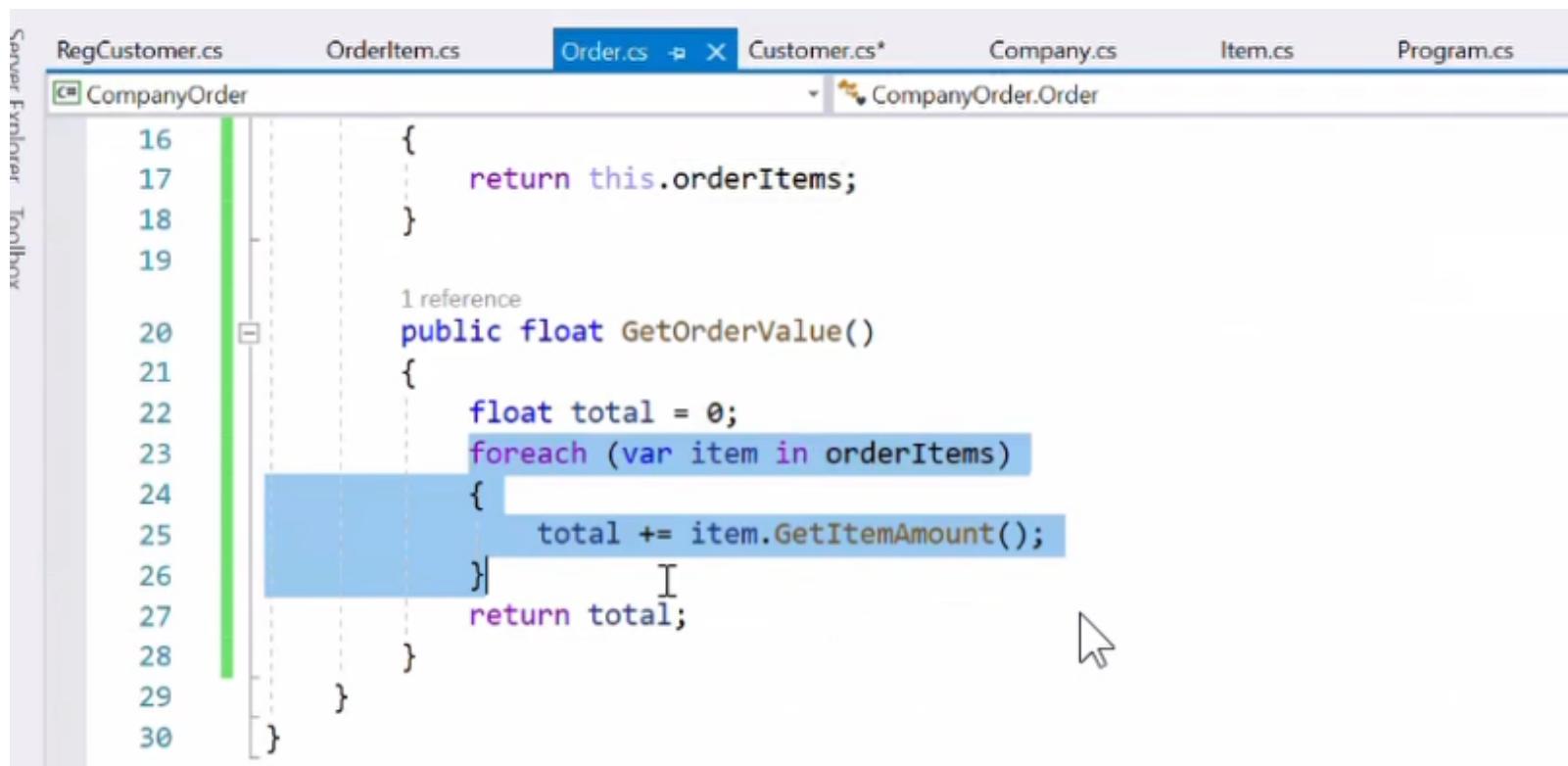
```
1  using System.Collections.Generic;
2
3  namespace CompanyOrder
4  {
5      class Order
6      {
7          public Customer Customer { get; set; }
8          private List<OrderItem> orderItems = new List<OrderItem>();
9
10         public void AddOrderItem(OrderItem orderItem)
11         {
12             this.orderItems.Add(orderItem);
13         }
14
15         public List<OrderItem> GetOrderItems()
16         {
17             return this.orderItems;
18         }
19
20         public float GetOrderValue()
```

Annotations from a static code analysis tool are present: a green vertical bar highlights the entire code block, and several code blocks are outlined with gray boxes and labeled with reference counts:

- Line 3: "16 references" (namespace CompanyOrder)
- Line 5: "6 references" (class Order)
- Line 8: "1 reference" (private List<OrderItem> orderItems = new List<OrderItem>();)
- Line 10: "9 references" (public void AddOrderItem(OrderItem orderItem))
- Line 15: "1 reference" (public List<OrderItem> GetOrderItems())
- Line 20: "1 reference" (public float GetOrderValue())

At the bottom left, it says "121 %". At the bottom center, there is a green checkmark icon and the text "No issues found".

we are looping through all those orderitem



```
16     {
17         return this.orderItems;
18     }
19
20     public float GetOrderValue()
21     {
22         float total = 0;
23         foreach (var item in orderItems)
24         {
25             total += item.GetItemAmount();
26         }
27         return total;
28     }
29 }
30 }
```

and we are calling GetItemAmount

```
RegCustomer.cs OrderItem.cs < X Order.cs Customer.cs* Company.cs Item.cs Program.cs  
CompanyOrder  
1  namespace CompanyOrder  
2 {  
3     class OrderItem  
4     {  
5         public int Qty { get; set; }  
6         public Item Item { get; set; }  
7  
8         public float GetItemAmount()  
9         {  
10             return Qty * Item.Price;  
11         }  
12     }  
13 }
```

so that i will add in the ordervalue  
and i'm printing it here

The screenshot shows a Microsoft Visual Studio code editor window. The tabs at the top include RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs\*, Company.cs, Item.cs, Program.cs (which is the active tab), and DisplayCompanyD. The code in Program.cs is as follows:

```
118     DrawLine();
119     Console.WriteLine("Item Name\tPrice\tQty\tAmount");
120     DrawLine();
121     foreach (var order in customer.GetOrders())
122     {
123         foreach (var orderItem in order.GetOrderItems())
124         {
125             Console.WriteLine(orderItem.Item.ItemName + "\t\t" +
126                             orderItem.Item.Price + "\t" +
127                             orderItem.Qty + "\t" +
128                             orderItem.GetItemAmount());
129             //totalAmount += orderItem.GetItemAmount();
130         }
131         DrawLine();
132         Console.WriteLine("Total Amount : \t\t\t" + order.GetOrderValue());
133         DrawLine();
134         customerTotalAmount += totalAmount;
135         totalAmount = 0;
136     }
137     DrawLine();
138     if(customer is RegCustomer)
139     {
```

So that's become the order value

my order will have many orderitem, 1 order will have many order item

that's a reason we have loop through all those orderitem and we got the totalamount

Now my ques is I want to calculate for the customer, how to get the totalcustomer value

The screenshot shows a code editor with several tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs\*, Company.cs. The Customer.cs\* tab is active. Below the tabs, there is a status bar showing 'CompanyOrder.Customer'. The main area contains the following C# code:

```
13     this.orders.Add(order);
14 }
15
16     1 reference
17     public List<Order> GetOrders()
18     {
19         return this.orders;
20     }
21     0 references
22     public float GetTotalCustomerValue()
23     {
24     }
25 }
26 }
```

A cursor is positioned over the method name 'GetTotalCustomerValue()' in line 22. The code editor interface includes a vertical scrollbar on the left and a horizontal scrollbar at the bottom.

customer has what

----list of orders

RegCustomer.cs OrderItem.cs Order.cs Customer.cs\* X Company.cs Item.cs

CompanyOrder

Customer.cs\* CompanyOrder.Customer

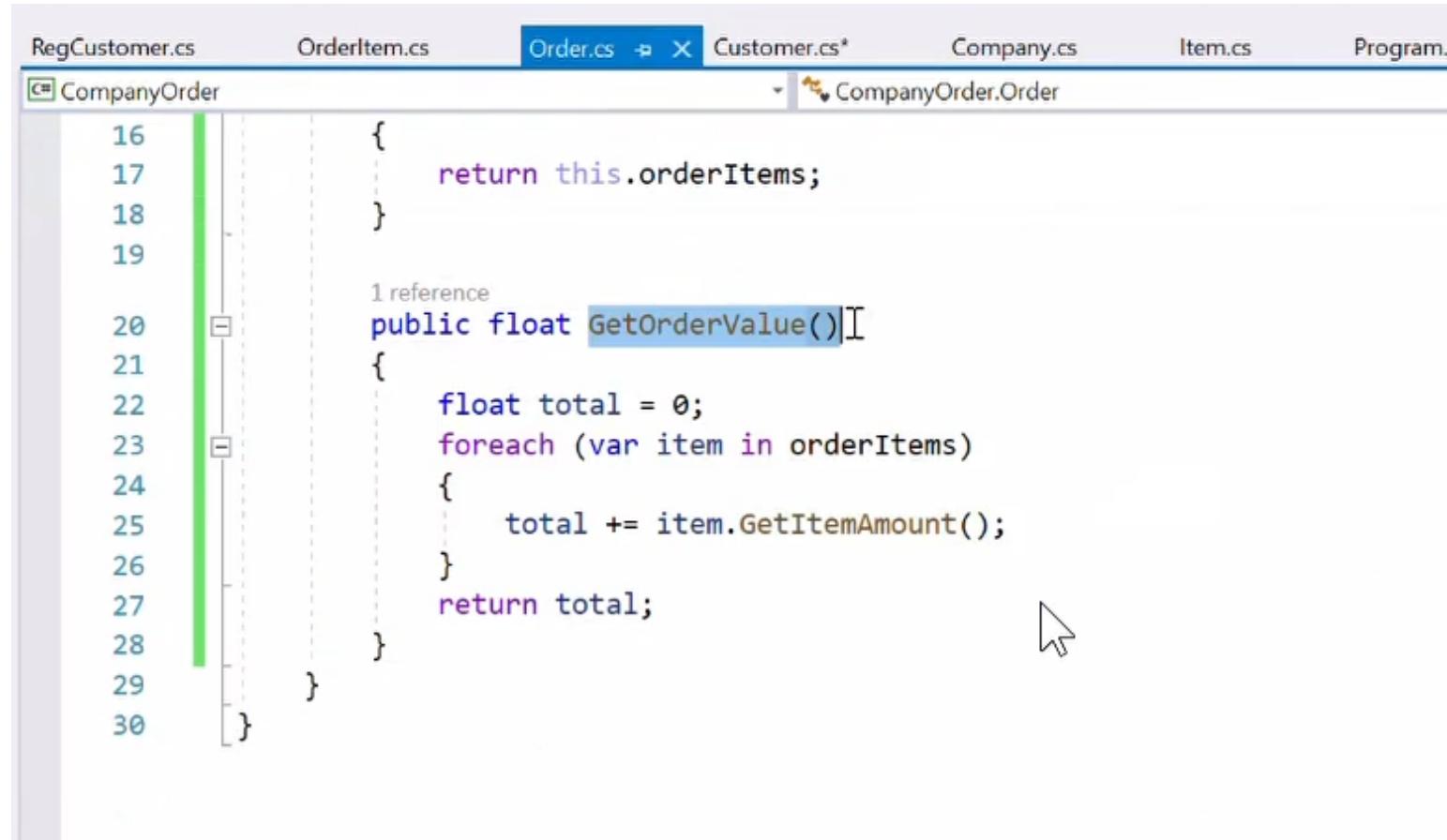
```
4 {  
5     11 references  
6     class Customer  
7     {  
8         4 references  
9         public string CustomerName { get; set; }  
10        3 references  
11        public Company Company { get; set; }  
12        private List<Order> orders = new List<Order>();  
13        6 references  
14        public void AddOrder(Order order)  
15        {  
16            this.orders.Add(order);  
17        }  
18        1 reference  
19        public List<Order> GetOrders()  
20        {  
21            return this.orders;  
22        }  
23        0 references  
24        public float GetTotalCustomerValue()  
25        {  
26    }
```

121 % 1 0 ← → | ⌂ ▾

Output Package Manager Console Error List Immediate Window

and each order has what

----GetOrderValue



The screenshot shows a code editor with multiple tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs (selected), Customer.cs\*, Company.cs, Item.cs, and Program.cs. The main pane displays the Order.cs file. A green vertical bar highlights the code from line 16 to line 29. The GetOrderValue() method is highlighted with a blue selection. A cursor arrow is positioned near the end of the method's body.

```
16     {
17         return this.orderItems;
18     }
19
20     public float GetOrderValue()
21     {
22         float total = 0;
23         foreach (var item in orderItems)
24         {
25             total += item.GetItemAmount();
26         }
27         return total;
28     }
29 }
30 }
```

when i say GetOrderValue i'll get this orderamount isn't it

# Shopping Cart

[Deselect all items](#)

Price



Mi 11X 5G Cosmic Black 6GB RAM 128GB ROM | SD 870 |  
DisplayMate A+ rated E4 AMOLED | Upto 18 Months No C...

₹27,999.00

In stock

Eligible for FREE Shipping

Fulfilled

This will be a gift [Learn more](#)

Colour: Cosmic Black

Style name: 6GB RAM 128GB Storage

Qty: 1  [Delete](#) [Save for later](#) [See more like this](#)



Samsung Galaxy M12 (Blue, 4GB RAM, 64GB Storage) 6000  
mAh with 8nm Processor | True 48 MP Quad Camera | 90H...

₹9,499.00

#1 Best Seller in Electronics

In stock

Eligible for FREE Shipping

Fulfilled

This will be a gift [Learn more](#)

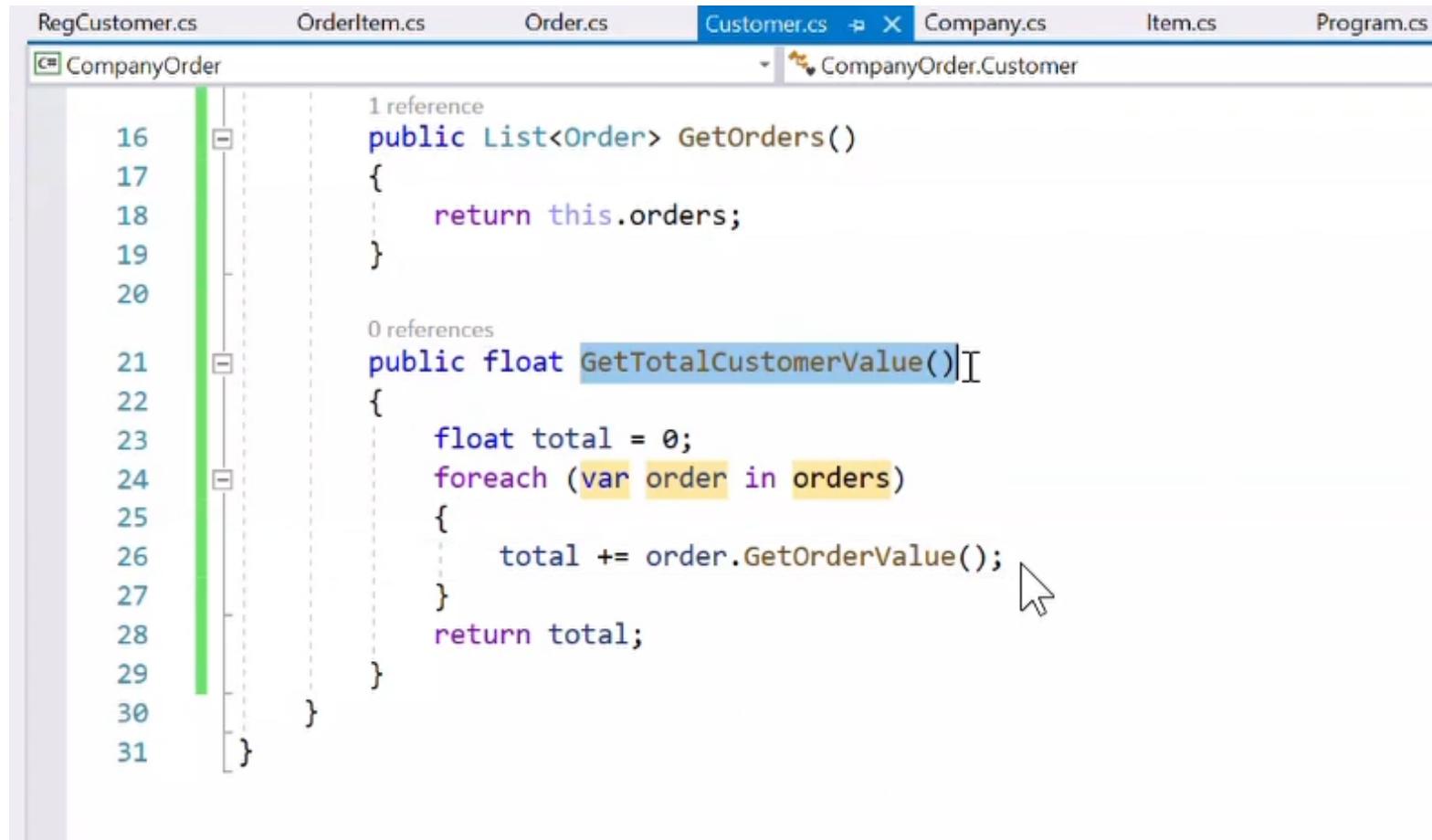
Colour: Blue

Size name: 4GB RAM & 64GB Storage

Qty: 2  [Delete](#) [Save for later](#) [See more like this](#)

Subtotal (3 items): ₹46,997.00

So for every order i'll get this ordervalue  
so that's what we'll do we'll loop through



The screenshot shows a code editor with multiple tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, and Program.cs. The Customer.cs tab is active. Below the tabs, there is a breadcrumb navigation bar with 'CompanyOrder' and 'CompanyOrder.Customer'. The main code area displays two methods:

```
16     1 reference
17     public List<Order> GetOrders()
18     {
19         return this.orders;
20     }
21     0 references
22     public float GetTotalCustomerValue()
23     {
24         float total = 0;
25         foreach (var order in orders)
26         {
27             total += order.GetOrderValue();
28         }
29     }
30 }
31 }
```

A mouse cursor is visible over the closing brace of the GetTotalCustomerValue method.

so this is at the customer level, so customer has a list of order, we are looping through the order, so each order will have a GetItemAmount

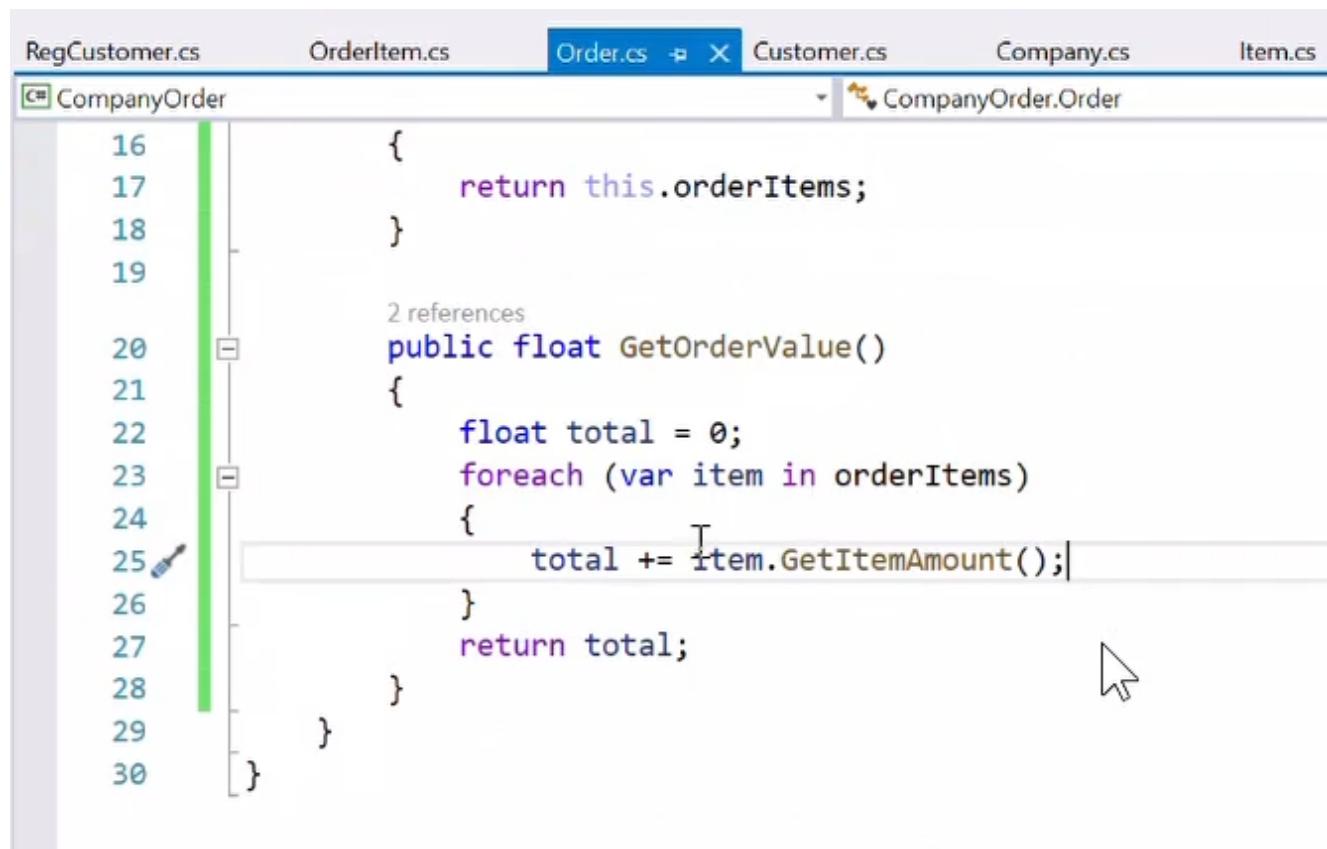
So each orderitem has GetItemAmount so that's get calculated

The screenshot shows a code editor with multiple tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs (selected), Customer.cs, and Company. The Order.cs tab has a tooltip 'CompanyOrder.Order' over it. The code in the editor is:

```
16     {
17         return this.orderItems;
18     }
19
20     public float GetOrderValue()
21     {
22         float total = 0;
23         foreach (var item in orderItems)
24         {
25             total += item.GetItemAmount(); // Line 25
26         }
27         return total;
28     }
29 }
30 }
```

A green vertical bar highlights the entire code block. A red bracket on the left side of the code indicates the scope of the 'orderItems' variable.

and that will be returned back to this



```
RegCustomer.cs OrderItem.cs Order.cs < X Customer.cs Company.cs Item.cs
CompanyOrder.cs CompanyOrder.Order
16     {
17         return this.orderItems;
18     }
19
20     public float GetOrderValue()
21     {
22         float total = 0;
23         foreach (var item in orderItems)
24         {
25             total += item.GetItemAmount();
26         }
27         return total;
28     }
29 }
30 }
```

and this gets calculated and returned back to this particular value

The screenshot shows a code editor with several tabs at the top: RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs (selected), Company.cs, and Item.cs. The current file is CompanyOrder.cs, which contains the following code:

```
16     1 reference
17     public List<Order> GetOrders()
18     {
19         return this.orders;
20     }
21     0 references
22     public float GetTotalCustomerValue()
23     {
24         float total = 0;
25         foreach (var order in orders)
26         {
27             total += order.GetOrderValue();
28         }
29     }
30 }
31 }
```

A green vertical bar highlights the entire class definition. A cursor is positioned over the closing brace of the `GetTotalCustomerValue()` method on line 31.

and that's what we'll print it here

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Project Explorer:** Shows files like RegCustomer.cs, OrderItem.cs, Order.cs, Customer.cs, Company.cs, Item.cs, and Program.cs.
- Toolbars:** Standard toolbar with icons for New, Open, Save, Cut, Copy, Paste, Find, and others.
- Status Bar:** Displays "121 %", "0", "2", "Ln: 138".
- Output Window:** Shows "Output", "Package Manager Console", "Error List ...", and "Immediate Window".

The **Program.cs** file contains the following C# code:

```
RegCustomer.cs OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs X
CompanyOrder.cs CompanyOrder.Program DisplayCompanyDetails(Company com)
130 }
131 DrawLine();
132 Console.WriteLine("Total Amount : \t\t\t" + order.GetOrderValue());
133 DrawLine();
134 //customerTotalAmount += totalAmount;
135 totalAmount = 0;
136 }
137 DrawLine();
138 Console.WriteLine("Total Customer Value : \t" + customer.GetTotalCustomerValue());
139 ▲ 11 of 19 ▼ void Console.WriteLine(string value)
    Writes the specified string value, followed by the current line terminator, to the standard output stream.
    value: The value to write.
140
141
142     RegCustomer regCustomer = (RegCustomer)customer;
143     int discount = regCustomer.Discount;
144     float discountTotal = customer.TotalAmount - customerTotalAmount * discount / 100;
145     Console.WriteLine("Total Customer Value : \t" + discountTotal);
146 }
147 else
148 {
149     Console.WriteLine("Total Customer Value : \t" + customerTotalAmount);
150 }
/*
151
152 customerTotalAmount = 0;
153 DrawLine();
```

A tooltip for the `Console.WriteLine(string value)` method is displayed, stating: "Writes the specified string value, followed by the current line terminator, to the standard output stream." The parameter `value` is described as "The value to write."

```
st: C:\WINDOWS\system32\cmd.exe
np
          Company Details
1-----
1:Company Name Samsung
1-----
1:Customer Name :Rajesh
1-----
1:Item Name      Price   Qty    Amount
1-----
1:Note1          23000   1      23000
1-----
1:Total Amount :           23000
1-----
1:Note1          23000   2      46000
1:Note4          28000   1      28000
1-----
1:Total Amount :           74000
ou
d
-- Total Customer Value : 97000
\S
\sCustomer Name :Ravi
cc
--Item Name      Price   Qty    Amount
--Note5          25000   4      100000
--Note1          23000   3      69000
--Total Amount :           169000
--Note3          26000   2      52000
```

Now ho do we calculate for registered customer  
map this with what we have learned yesterday team it will be easy for u

my animal class has one method

```
RegCustomer.cs      OrderItem.cs      Order.cs      Customer.cs      Company.cs      Item.cs      Program.cs
CompanyOrder
Customer.cs  X  Company.cs
CompanyOrder.Customer

16     1 reference
17         public List<Order> GetOrders()
18             {
19                 return this.orders;
20             }
21     1 reference
22         public float GetTotalCustomerValue()
23             {
24                 float total = 0;
25                 foreach (var order in orders)
26                 {
27                     total += order.GetOrderValue();
28                 }
29             }
30         }
31     }
```

and i want to have same implementation in the child class also  
but the implementation should be different

So that when i say animal.walk [animal.eat](#) it should go to child eat, when i pass dog object it should call dog's eat method not animal's eat method

Now what happening is it is passing me common information which is comming from animal class and i want to write a specific code to that particular dog class so how will you make sure, the dog's information is coming not the animals  
how will i do that

----overriding

we need to override it

we need to write the same method what we have here

```
RegCustomer.cs* OrderItem.cs Order.cs Customer.cs X Company.cs Item.cs Program.cs  
C# CompanyOrder  
1 reference  
16     public List<Order> GetOrders()  
17     {  
18         return this.orders;  
19     }  
20  
1 reference  
21     public float GetTotalCustomerValue()  
22     {  
23         float total = 0;  
24         foreach (var order in orders)  
25         {  
26             total += order.GetOrderValue();  
27         }  
28         return total;  
29     }  
30     }  
31 }
```

i'll write the same method here

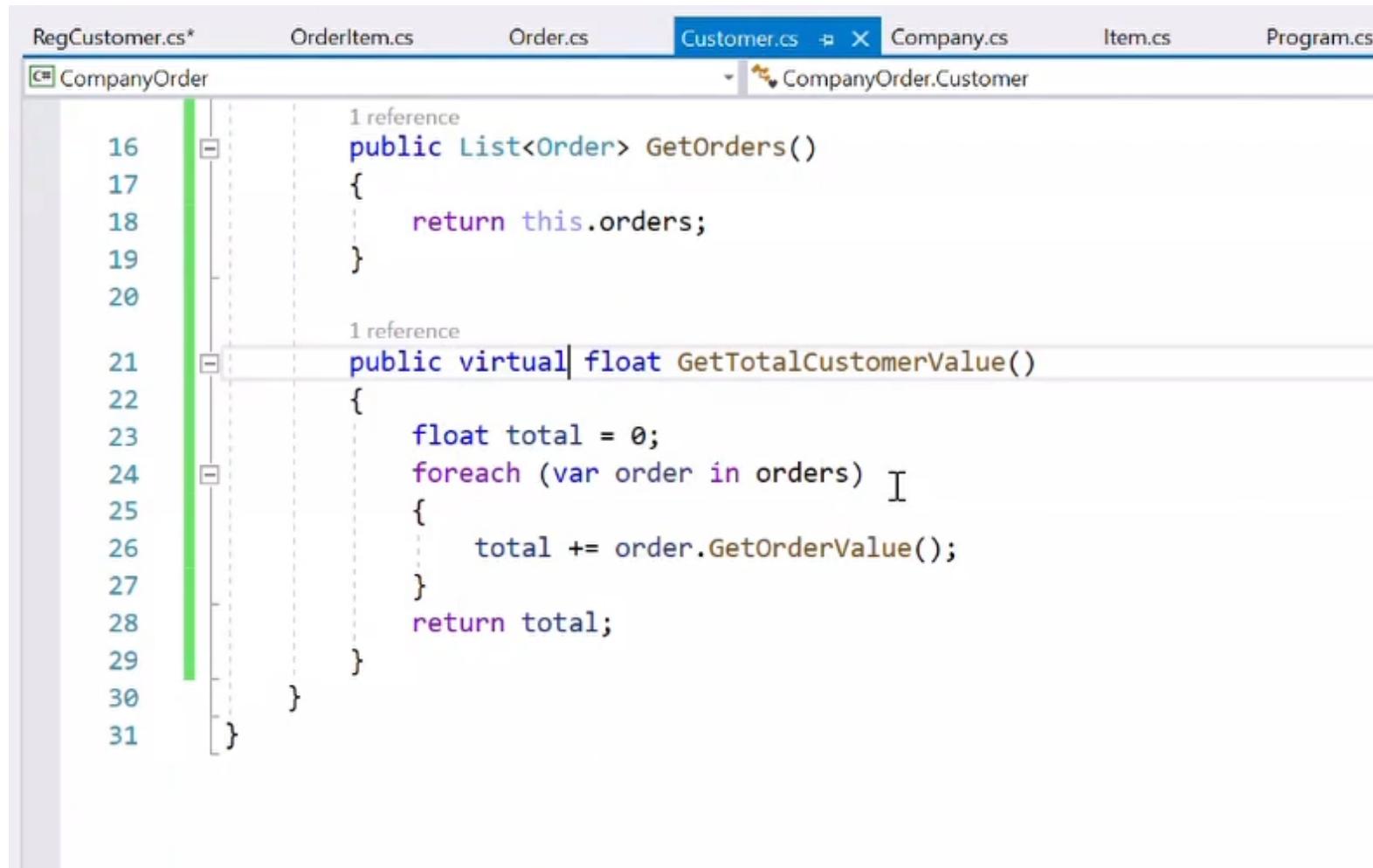
RegCustomer.cs\* OrderItem.cs Order.cs Customer.cs Company.cs Item.cs Program.cs

CompanyOrder CompanyOrder.RegCustomer

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace CompanyOrder
8  {
9      class RegCustomer : Customer
10     {
11         public int Discout { get; set; } = 10;
12
13         public float GetTotalCustomerValue()
14         {
15             float total = 0;
16             foreach (var order in orders)
17             {
18                 total += order.GetOrderValue();
19             }
20             return total;
21         }
22     }
23 }
```

121 % No issues found

but when i'm writing this  
how will you say that this is overwritten here  
---in the base class you should say virtual



The screenshot shows a code editor with multiple tabs at the top: RegCustomer.cs\*, OrderItem.cs, Order.cs, Customer.cs (selected), Company.cs, Item.cs, and Program.cs. The main pane displays the code for the Customer.cs file. The code includes two methods: GetOrders() and GetTotalCustomerValue(). The GetTotalCustomerValue() method is annotated with '1 reference' and 'public virtual float GetTotalCustomerValue()'.

```
16     public List<Order> GetOrders()
17     {
18         return this.orders;
19     }
20
21     public virtual float GetTotalCustomerValue()
22     {
23         float total = 0;
24         foreach (var order in orders)
25         {
26             total += order.GetOrderValue();
27         }
28         return total;
29     }
30     }
31 }
```

and in child we have to say override

```
RegCustomer.cs* OrderItem.cs Order.cs Customer.cs Company.cs
CompanyOrder
7  namespace CompanyOrder
8  {
9      class RegCustomer : Customer
10     {
11         public int Discout { get; set; } = 10;
12
13         public override float GetTotalCustomerValue()
14         {
15             float total = 0;
16             foreach (var order in orders)
17             {
18                 total += order.GetOrderValue();
19             }
20             return total;
21         }
22     }
23 }
24
```

now we get the error, from the child class can i access orders  
it cannot because orders is private

RegCustomer.cs\* OrderItem.cs Order.cs Customer.cs X Company.cs Item.cs

CompanyOrder

using System.Collections.Generic;

namespace CompanyOrder

{

class Customer

{

public string CustomerName { get; set; }

public Company Company { get; set; }

private List<Order> orders = new List<Order>();

public void AddOrder(Order order)

{

this.orders.Add(order);

}

public List<Order> GetOrders()

{

return this.orders;

}

No issues found

Output Package Manager Console Error List Immediate Window

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

so we should call this particular method