**TechShop, an electronic gadgets shop**

**Name:**Jeffry Joshuva Amalan J

**Implement OOPs**

**Task 1: Classes and Their Attributes:**

**Task 2: Class Creation:**

**Task 3: Encapsulation:**

Below I have completed the above 3 task combined in a single code as class creation with their attributes as well as making the required phase as encapsulation

using System;

using System.Collections;

using System.Collections.Generic;

using System.Collections.Specialized;

using System.Diagnostics;

using System.Linq;

using System.Net;

using System.Runtime.CompilerServices;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

public class Customers

{

private int CustomerID;

private string FirstName;

private string LastName;

private string Email;

private string Phone;

private string Address;

public int customerID

{

get { return CustomerID; }

set { CustomerID = value; }

}

public string Firstname

{

get { return FirstName; }

set { FirstName = value; }

}

public string Lastname

{

get { return LastName; }

set { LastName = value; }

}

public string email

{

get { return Email; }

set { Email = value; }

}

public string phone

{

get { return Phone; }

set { Phone = value; }

}

public string address

{

get { return Address; }

set { Address = value; }

}

public void CalculateTotalOrders()

{

//have to return the oders count;

}

public void GetCustomerDetails()

{

Console.WriteLine($"CustomerID: {CustomerID}, CustomerName: {FirstName} {LastName}, Email: {Email}, Phone: {Phone}, Address: {Address}");

}

public void UpdateCustomerInfo(string email, string phone, string address)

{

Email = email;

Phone = phone;

Address = address;

}

}

public class Products

{

private int ProductID;

public string ProductName;

private string Description;

private decimal Price;

public int productID

{

get { return ProductID; }

set { ProductID = value; }

}

public string Productname

{

get { return ProductName; }

set { ProductName = value; }

}

public string description

{

get { return Description; }

set { Description = value; }

}

public decimal price

{

get { return Price; }

set { Price = value < 0 ? 0 : value; }

}

public void GetProductDetails()

{

Console.WriteLine($"ID: {ProductID}, Name: {ProductName}, Description: {Description}, Price: {Price:C}");

}

public void UpdateProductInfo(decimal newPrice, string newDescription)

{

Price = newPrice;

Description = newDescription;

}

public void IsProductInStock()

{

//have to check whether the product is available or not;

}

}

public class Orders

{

private int OrderID;

private Customers customer;

private DateTime orderDate;

private decimal totalAmount;

private string status;

public int orderID

{

get { return OrderID; }

set { OrderID = value; }

}

public Customers Customer

{

get { return customer; }

set { customer = value; }

}

public DateTime OrderDate

{

get { return orderDate; }

set { orderDate = value; }

}

public decimal TotalAmount

{

get { return totalAmount; }

private set { totalAmount = value; }

}

public string Status

{

get { return status; }

private set { status = value; }

}

public void CalculateTotalAmount()

{

//to calculate total amount

}

public void GetOrderDetails()

{

Console.WriteLine($"OrderID: {OrderID}, Date: {OrderDate}, Status: {status}, Total: {totalAmount:C}");

}

public void UpdateOrderStatus(string newStatus)

{

status = newStatus;

}

public void CancelOrder()

{

status = "cancelled";

}

}

public class OrderDetails

{

private int orderDetailID;

private Orders order;

private Products product;

private int quantity;

private decimal discount;

public int OrderDetailID

{

get { return orderDetailID; }

set { orderDetailID = value; }

}

public Orders Order

{

get { return order; }

set { order = value; }

}

public Products Product

{

get { return product; }

set { product = value; }

}

public int Quantity

{

get { return quantity; }

set { quantity = value > 0 ? value : 1; }

}

public decimal CalculateSubtotal()

{

return (product.price \* quantity) \* (1 - discount);

}

public void GetOrderDetailInfo()

{

Console.WriteLine($"Product: {product.ProductName}, Qty: {Quantity}, Subtotal: {CalculateSubtotal():C}");

}

public void UpdateQuantity(int newQuantity)

{

Quantity = newQuantity;

}

public void AddDiscount(decimal discountPercent)

{

discount = discountPercent < 0 || discountPercent > 1 ? 0 : discountPercent;

}

}

public class Inventory

{

private int inventoryID;

private Products product;

private int quantityInStock;

private DateTime lastStockUpdate;

public int InventoryID

{

get { return inventoryID; }

set { inventoryID = value; }

}

public Products Product

{

get { return product; }

set { product = value; }

}

public int QuantityInStock

{

get { return quantityInStock; }

private set { quantityInStock = value < 0 ? 0 : value; }

}

public DateTime LastStockUpdate

{

get { return lastStockUpdate; }

private set { lastStockUpdate = value; }

}

public void GetProduct()

{

//want to get the product

}

public void GetQuantityInStock()

{

//havro get the quantity in stock

}

public void AddToInventory(int quantity)

{

QuantityInStock += quantity;

LastStockUpdate = DateTime.Now;

}

public void RemoveFromInventory(int quantity)

{

QuantityInStock = Math.Max(QuantityInStock - quantity, 0);

LastStockUpdate = DateTime.Now;

}

public void UpdateStockQuantity(int newQuantity)

{

QuantityInStock = newQuantity;

}

public void IsProductAvailable(int quantityToCheck)

{

//have to check the product is available

}

public decimal GetInventoryValue() {

return product.price\* quantityInStock;

}

public void ListLowStockProducts(int threshold)

{

//have to list the product in stock

}

public void ListOutOfStockProducts()

{

//have to list the product thatis out of stock

}

public void ListAllProducts()

{

//have to list all the list of products

}

}

internal class Techshop

{

static void Main(string[] args)

{

// Create customer

Customers customer = new Customers

{

customerID = 1,

Firstname = "Jeffry",

Lastname = "Amalan",

email = "jeffry@example.com",

phone = "9876543210",

address = "Theni"

};

// Create products

Products laptop = new Products

{

productID = 101,

Productname = "Laptop",

description = "Gaming Laptop",

price = 75000

};

Products mouse = new Products

{

productID = 102,

Productname = "Wireless Mouse",

description = "Bluetooth Mouse",

price = 1500

};

// Add inventory

Inventory inv1 = new Inventory

{

InventoryID = 1,

Product = laptop

};

inv1.AddToInventory(10);

Inventory inv2 = new Inventory

{

InventoryID = 2,

Product = mouse

};

inv2.AddToInventory(20);

// Create order

Orders order = new Orders

{

orderID = 501,

Customer = customer,

OrderDate = DateTime.Now

};

// Add order details

OrderDetails detail1 = new OrderDetails

{

OrderDetailID = 1,

Order = order,

Product = laptop,

Quantity = 1

};

OrderDetails detail2 = new OrderDetails

{

OrderDetailID = 2,

Order = order,

Product = mouse,

Quantity = 2

};

detail2.AddDiscount(0.1m); // 10% discount

// Add details to order

order.CalculateTotalAmount(); // Not adding list but logic is there

// Print customer info

Console.WriteLine("customer details");

customer.GetCustomerDetails();

// Print order info

Console.WriteLine("\norder details");

order.GetOrderDetails();

// Print each order detail

Console.WriteLine("\norder line details");

detail1.GetOrderDetailInfo();

detail2.GetOrderDetailInfo();

// Print inventory value

Console.WriteLine($"\nInventory Value - Laptop: {inv1.GetInventoryValue():C}");

Console.WriteLine($"Inventory Value - Mouse: {inv2.GetInventoryValue():C}");

Console.ReadLine();

}

}

}