**C# Evaluation Test – 9 April, 2025**

**1)**

**namespace Task1**

**{**

**class Product**

**{**

**private int productid;**

**private string productName;**

**private string description;**

**private double price;**

**private int quantityinstock;**

**private string type;**

**}**

**}**

**2)**

**using System;**

**namespace Task2**

**{**

**public class Product**

**{**

**private int productid;**

**private string productName;**

**private string description;**

**private double price;**

**private int quantityinstock;**

**private string type;**

**public Product()**

**{**

**productid=0;**

**productName="None";**

**description="None";**

**price=0;**

**quantityinstock=0;**

**type="None";**

**}**

**public Product(int pid, string pname, string desc, double pri, int qis, string t)**

**{**

**productid=pid;**

**productName=pname;**

**description=desc;**

**price=pri;**

**quantityinstock=qis;**

**type=t;**

**}**

**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 double Price**

**{**

**get { return price; }**

**set { price=value; }**

**}**

**public int Quantityinstock**

**{**

**get { return quantityinstock; }**

**set { quantityinstock=value; }**

**}**

**public string Type**

**{**

**get { return type; }**

**set { type=value; }**

**}**

**public virtual void PrintInfo()**

**{**

**Console.WriteLine("Product ID:"+productid);**

**Console.WriteLine("Product Name:"+productName);**

**Console.WriteLine("Description:"+description);**

**Console.WriteLine("Price:"+price);**

**Console.WriteLine("Quantity in Stock:"+quantityinstock);**

**Console.WriteLine("Type:"+type);**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

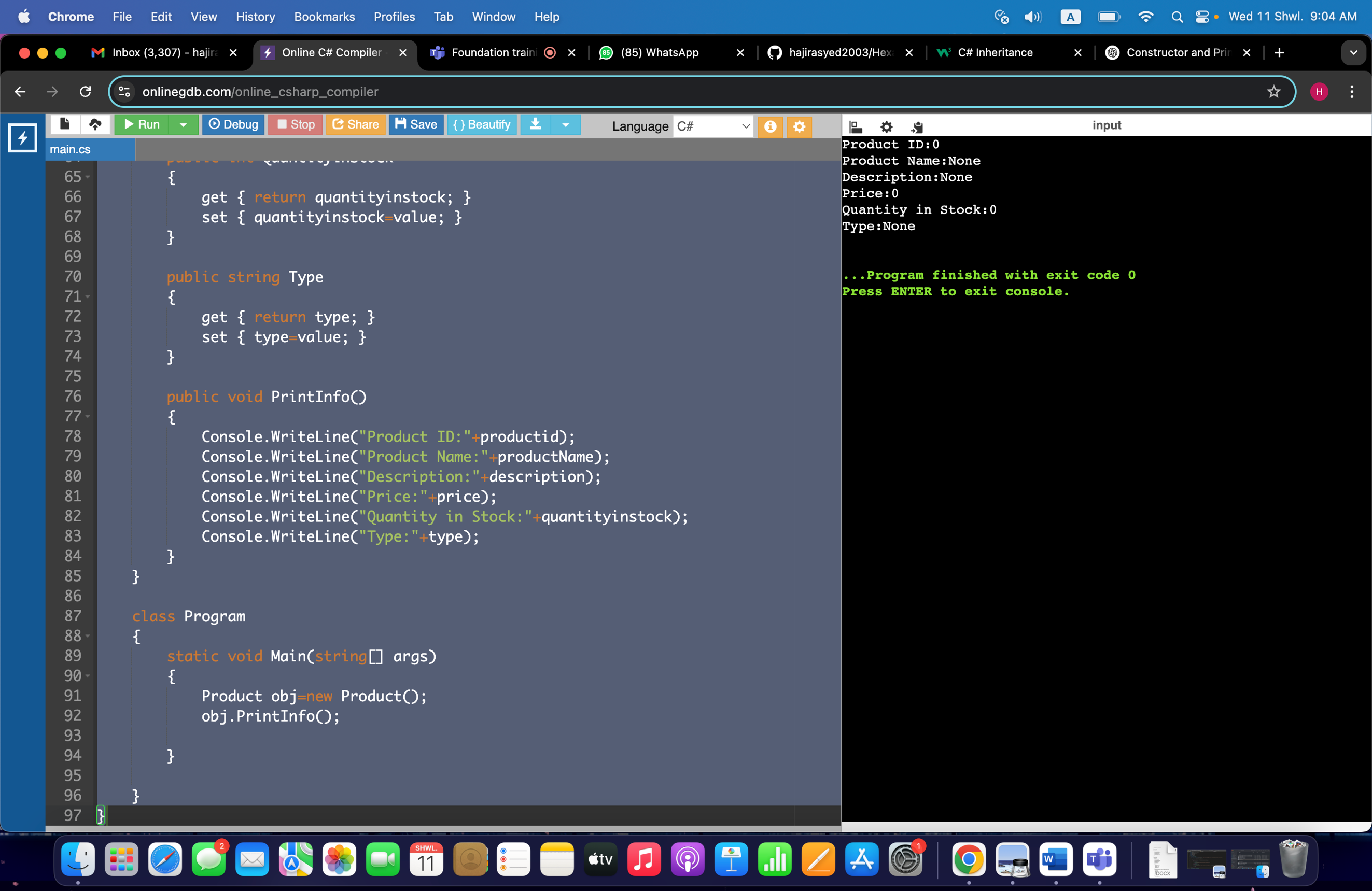
**Product obj=new Product();**

**obj.PrintInfo();**

**}**

**}**

**}**

****

**3)**

**class Electronics: Product**

**{**

**public string brand;**

**public int warrantyperiod;**

**public Electronics()**

**{**

**brand="None";**

**warrantyperiod=0;**

**}**

**public Electronics(string b, int w)**

**{**

**brand=b;**

**warrantyperiod=w;**

**}**

**public override void PrintInfo()**

**{**

**Console.WriteLine("Product ID:"+Productid);**

**Console.WriteLine("Product Name:"+ProductName);**

**Console.WriteLine("Description:"+Description);**

**Console.WriteLine("Price:"+Price);**

**Console.WriteLine("Quantity in Stock:"+Quantityinstock);**

**Console.WriteLine("Type:"+Type);**

**Console.WriteLine("Brand:"+brand);**

**Console.WriteLine("Warranty Period:"+warrantyperiod);**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

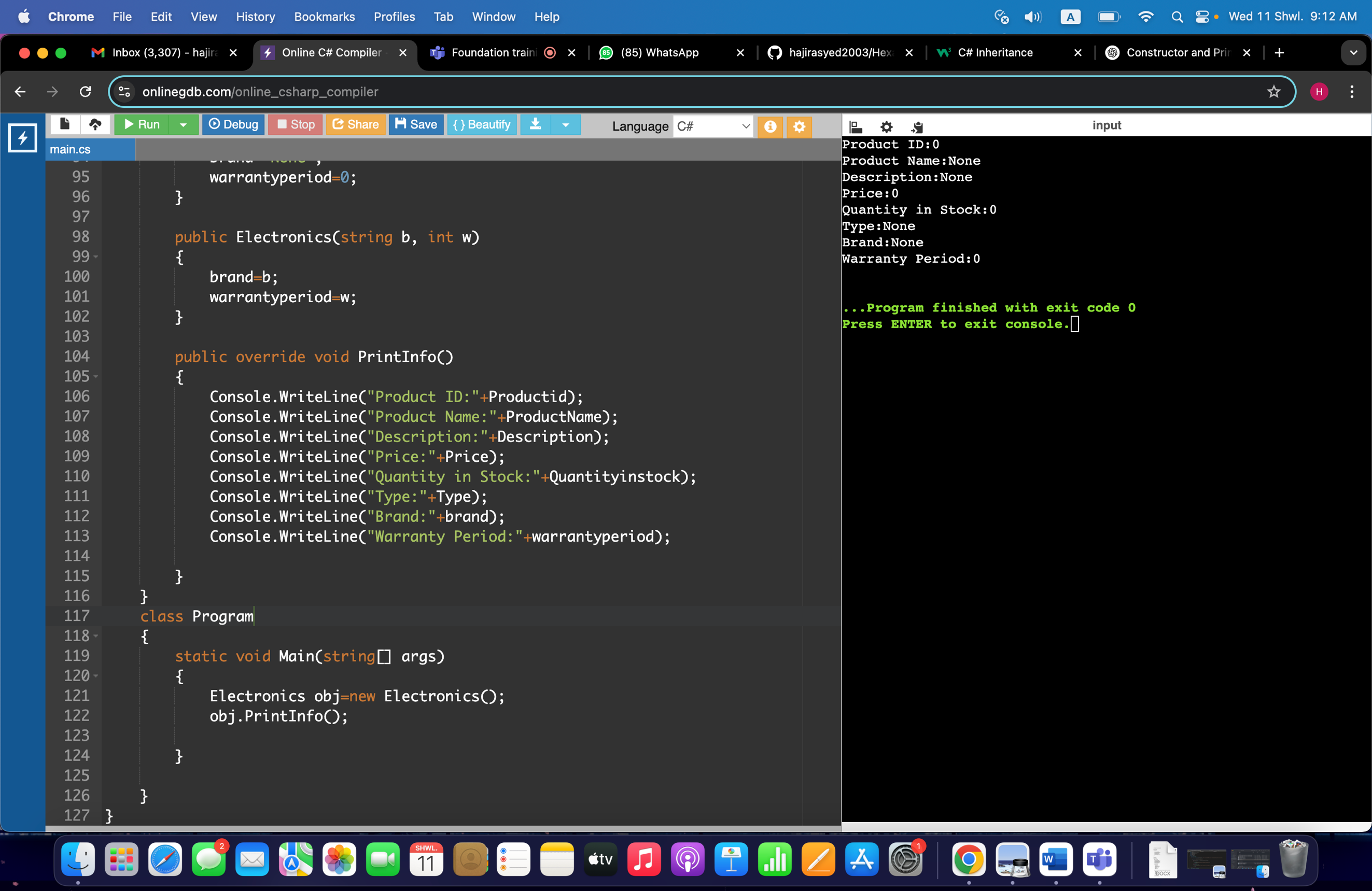
**{**

**Electronics obj=new Electronics();**

**obj.PrintInfo();**

**}**

**}**

****

**4)**

**class Clothing : Product**

**{**

**public string size;**

**public string color;**

**public Clothing()**

**{**

**size="None";**

**color="None";**

**}**

**public Clothing(string s, string c)**

**{**

**size=s;**

**color=c;**

**}**

**public override void PrintInfo()**

**{**

**Console.WriteLine("Product ID:"+Productid);**

**Console.WriteLine("Product Name:"+ProductName);**

**Console.WriteLine("Description:"+Description);**

**Console.WriteLine("Price:"+Price);**

**Console.WriteLine("Quantity in Stock:"+Quantityinstock);**

**Console.WriteLine("Type:"+Type);**

**Console.WriteLine("Size:"+size);**

**Console.WriteLine("Color:"+color);**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

**{**

**Console.WriteLine("Enter the size and color of the Product:");**

**string Size=Console.ReadLine();**

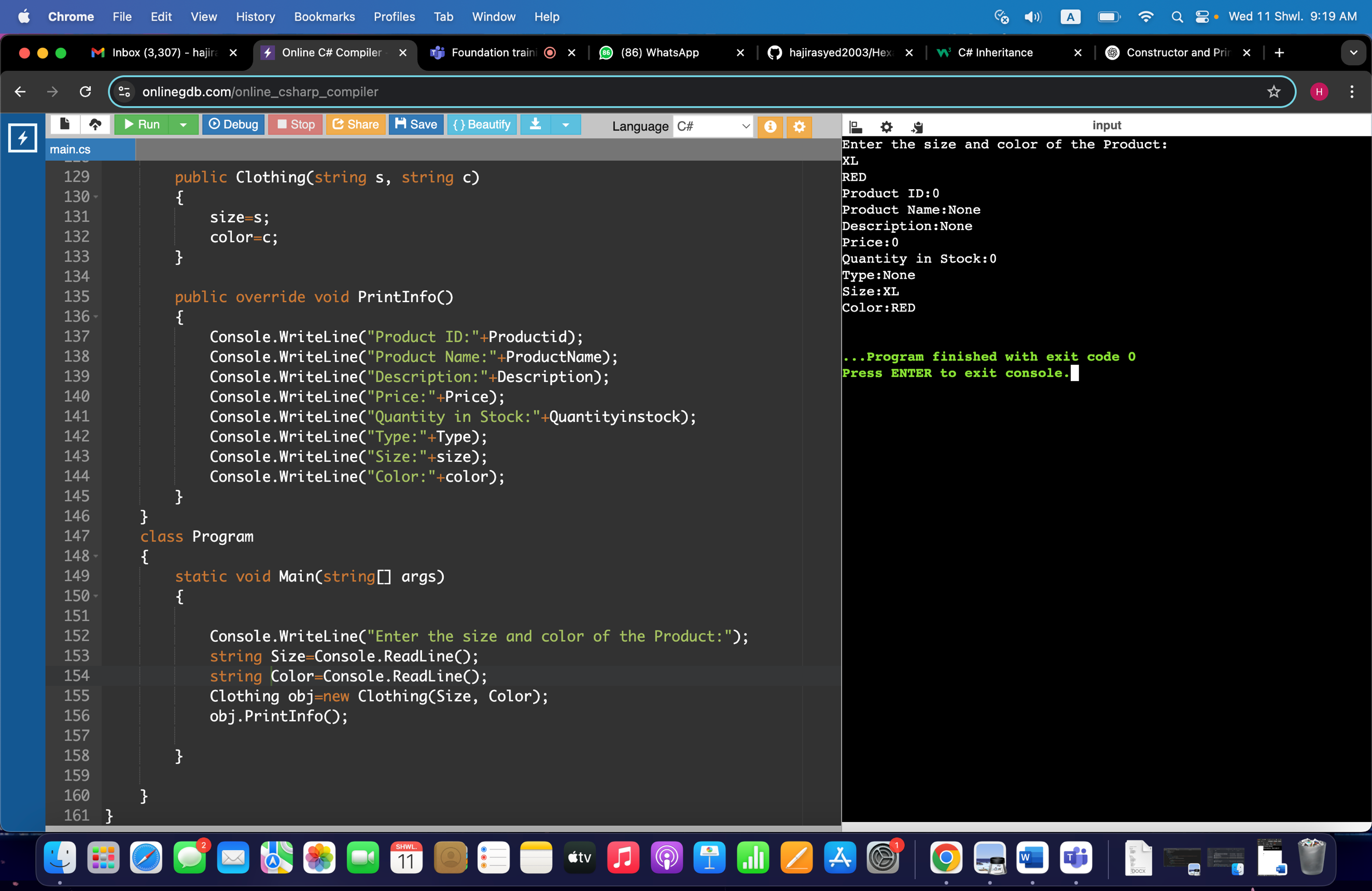
**string Color=Console.ReadLine();**

**Clothing obj=new Clothing(Size, Color);**

**obj.PrintInfo();**

**}**

**}**

****

**5)**

**class User**

**{**

**public int userId;**

**public string username;**

**public string password;**

**public string role;**

**public User()**

**{**

**userId=0;**

**username="None";**

**password="None";**

**role="None";**

**}**

**public User(int uid, string uname, string pass, string rol)**

**{**

**userId=uid;**

**username=uname;**

**password=pass;**

**role=rol;**

**}**

**public void PrintInfo2()**

**{**

**Console.WriteLine("User Id:"+userId);**

**Console.WriteLine("Username:"+username);**

**Console.WriteLine("Password:"+password);**

**Console.WriteLine("Role (Admin/User) :"+role);**

**}**

**}**

**class Program**

**{**

**static void Main(string[] args)**

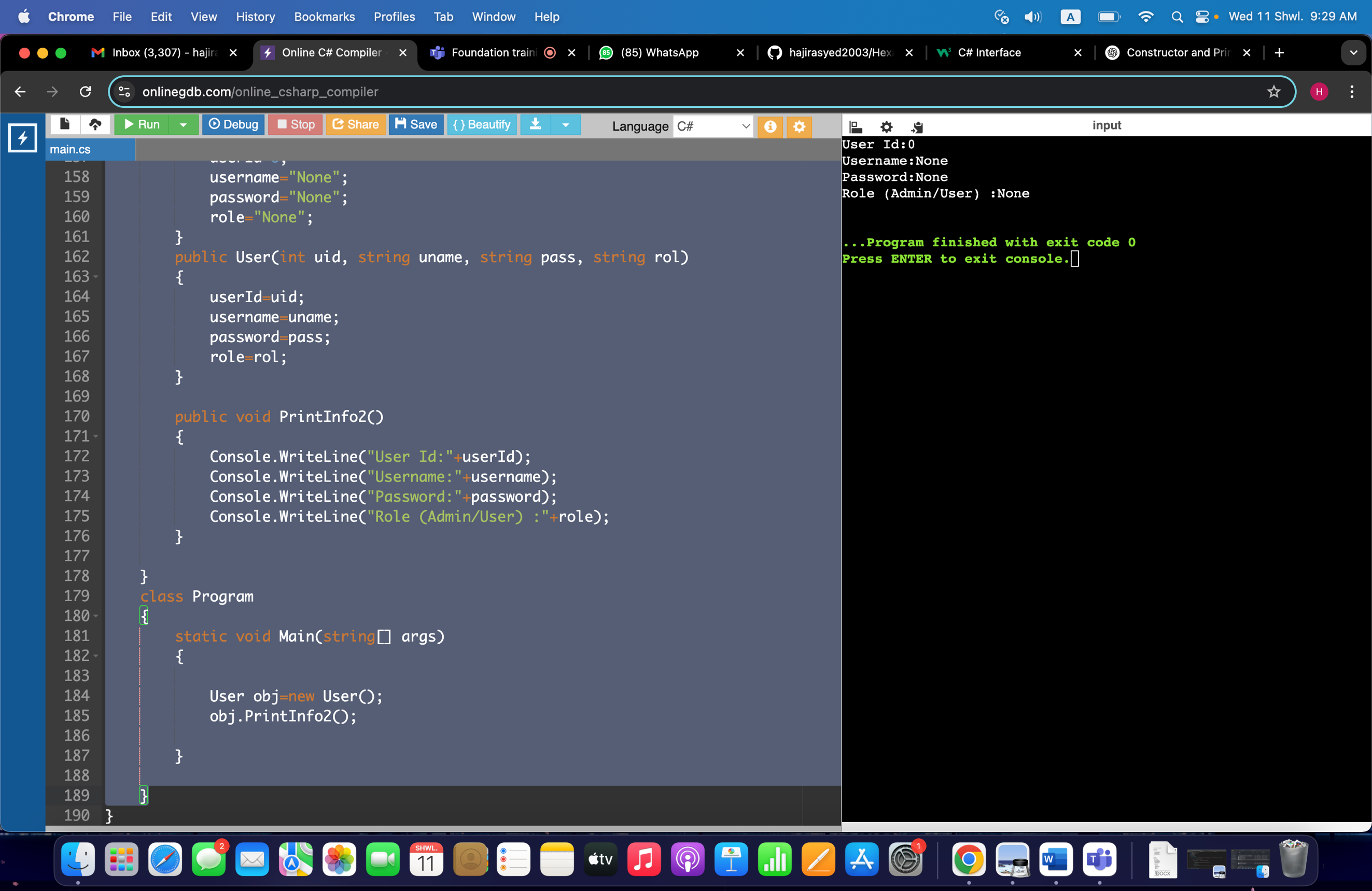
**{**

**User obj=new User();**

**obj.PrintInfo2();**

**}**

**}**

****

Since an additional order class is required, an order class created:

class Order

{

public int orderId;

public int userId;

public List<Product> products;

public Order()

{

orderId = 0;

userId = 0;

products = new List<Product>();

}

public Order(int oid, int uid, List<Product> prodList)

{

orderId = oid;

userId = uid;

products = prodList;

}

}

**6)**

**Using System;**

**using System.Collections.Generic;**

**public interface IOrderManagementRepository**

**{**

**void createOrder(User user, List<Product> products);**

**void cancelOrder(int userId);**

**void createProduct(Product product);**

**void createUser(User user);**

**List<Product> GetAllProduct();**

**List<Product> GetOrderByUser(User user);**

**}**

**7)**

**using System;**

**using System.Collections.Generic;**

**namespace Task7**

**{**

**public class OrderProcessor : IOrderManagementRepository**

**{**

**private List<User> users = new List<User>();**

**private List<Product> products = new List<Product>();**

**private Dictionary<int, List<Product>> orders = new Dictionary<int, List<Product>>();**

**public void createOrder(User user, List<Product> productList)**

**{**

**User euser=null;**

**foreach (User u in users)**

**{**

**if (u.userId==user.userId)**

**{**

**euser=u;**

**break;**

**}**

**}**

**if (euser == null)**

**{**

**Console.WriteLine("Creating New User");**

**users.Add(user);**

**}**

**if (!orders.ContainsKey(user.userId))**

**{**

**orders[user.userId] = new List<Product>();**

**}**

**orders[user.userId].AddRange(productList);**

**Console.WriteLine("Order created");**

**}**

**public void cancelOrder(int userId)**

**{**

**if (!orders.ContainsKey(userId))**

**{**

**throw new UserNotFoundException("User does not exist");**

**}**

**orders[userId].RemoveAt(orderId - 1);**

**Console.WriteLine("Order cancelled successfully.");**

**}**

**public void createProduct(Product product)**

**{**

**products.Add(product);**

**Console.WriteLine("Product created successfully.");**

**}**

**public void createUser(User user)**

**{**

**users.Add(user);**

**Console.WriteLine("User added successfully.");**

**}**

**public List<Product> GetAllProduct()**

**{**

**return products;**

**}**

**public List<Product> GetOrderByUser(int userid)**

**{**

**return (orders.ContainsKey(userId)) ? orders[userId] : new List<Product>();**

**}**

**}**

**}**

**8)**

**using System.Data.SqlClient;**

**public class DBUtil**

**{**

**private static string connectionstring="Data Source=SQLEXPRESS;Initial Catalog=YOUR\_DATABASE;Integrated Security=True";**

**public static SqlConnection Conn()**

**{**

**SqlConnection conn=new SqlConnection(connectionstring);**

**try**

**{**

**conn.Open();**

**Console.WriteLine("SQL Connection has been successfully established.");**

**}**

**catch (Exception ex)**

**{**

**Console.WriteLine("Connection Error"+ex.message);**

**}**

**return conn;**

**}**

**}**

**9)**

**namespace Task9**

**{**

**class OrderManagement**

**{**

**static void Main(string[] args)**

**{**

**Console.WriteLine("Menu");**

**Console.WriteLine("1. Create User");**

**Console.WriteLine("2. Create Product");**

**Console.WriteLine("3. Cancel Order");**

**Console.WriteLine("4. Get All Products");**

**Console.WriteLine("5. Get Order by User");**

**Console.WriteLine("6. Exit");**

**Console.WriteLine("Enter valid choice");**

**int opt=Convert.ToInt32(Console.ReadLine());**

**OrderProcessor op= new OrderProcessor();**

**switch(opt)**

**{**

**case 1:**

**Console.WriteLine("Enter Userid, username, password, role"):**

**int id=Convert.ToInt32(Console.ReadLine());**

**string u=Console.ReadLine();**

**string p=Console.ReadLine();**

**string r=Console.ReadLine();**

**User u=new User(id,u,p,r);**

**op.createUser(u);**

**break;**

**case 2:**

**Console.WriteLine("Enter ProductId, Name, Description, Price, Qty, Type:");**

**int pid2=Convert.ToInt32(Console.ReadLine());**

**n2=Console.ReadLine();**

**d2=Console.ReadLine();**

**pr2=Convert.ToDouble(Console.ReadLine());**

**q2=Convert.ToInt32(Console.ReadLine());**

**t2=Console.ReadLine();**

**Product p=new Product(pid2,n2,d2,pr2,q2,t2);**

**op.createProduct(p);**

**break;**

**case 3:**

**Console.WriteLine("Enter Userid:");**

**int uid3=Convert.ToInt32(Console.ReadLine());**

**op.cancelOrder(uid3);**

**break;**

**case 4:**

**List p4= op.GetAllProduct();**

**foreach (i in p4)**

**{**

**i.PrintInfo();**

**}**

**break;**

**case 5:**

**Console.WriteLine("Enter Userid:");**

**uid5=Convert.ToInt32(Console.ReadLine());**

**List p5=op.GetOrderByUser(uid5);**

**foreach (j in p5)**

**{**

**j.PrintInfo();**

**}**

**break;**

**case 6:**

**return;**

**default:**

**Console.WriteLine("Enter valid choices");**

**break;**

**}**

**}**

**}**

**}**