**Exercise 1: Implementing the Singleton Pattern**

**Code:**

using System;

class Singleton

{

private static readonly Singleton instance = new Singleton();

private Singleton() { }

public static Singleton Instance

    {

        get { return instance; }

    }

public void ShowMessage()

    {

        Console.WriteLine("Hello from Sai!");

    }

}

class Program

{

    static void Main(string[] args)

    {

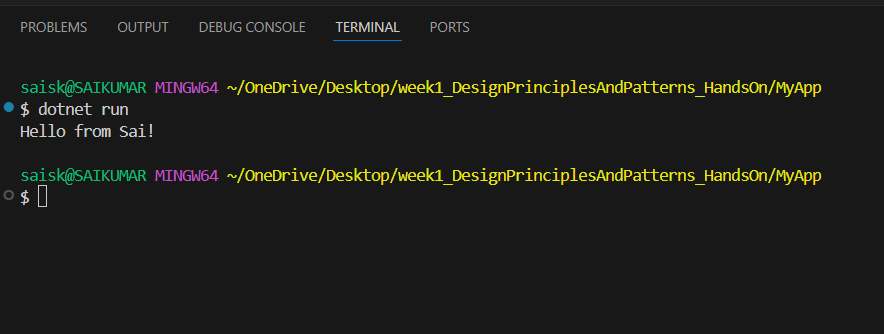
        Singleton obj = Singleton.Instance;

        obj.ShowMessage();

    }

}

**Output:**

****

**Exercise 2: Implementing the Factory Method Pattern**

**Code:**

using System;

public interface IVehicle

{

    void Drive();

}

public class Car : IVehicle

{

    public void Drive()

    {

        Console.WriteLine("Driving a car");

    }

}

public class Motorcycle : IVehicle

{

    public void Drive()

    {

        Console.WriteLine("Driving a motorcycle");

    }

}

public class VehicleFactory

{

    public IVehicle CreateVehicle(string type)

    {

        if (type.Equals("car", StringComparison.OrdinalIgnoreCase))

        {

            return new Car();

        }

        else if (type.Equals("motorcycle", StringComparison.OrdinalIgnoreCase))

        {

            return new Motorcycle();

        }

        return new Car();

    }

}

class Program

{

    static void Main()

    {

        VehicleFactory factory = new VehicleFactory();

        IVehicle car = factory.CreateVehicle("car");

        car.Drive();

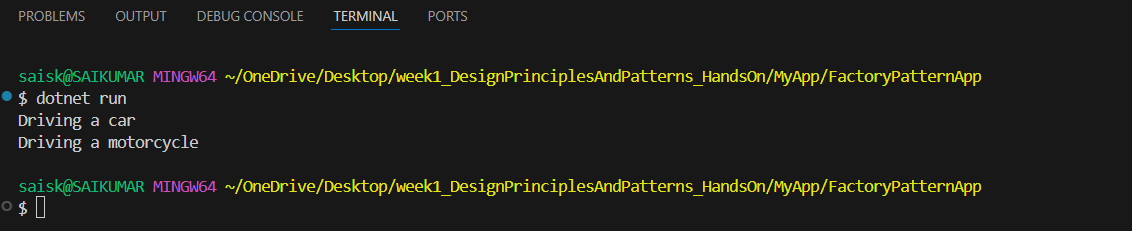
        IVehicle motorcycle = factory.CreateVehicle("motorcycle");

        motorcycle.Drive();

    }

}

**Output**

****

**Exercise 2: E-commerce Platform Search Function**

**Code:**

using System;

using System.Collections.Generic;

class Program

{

    static void Main()

    {

        List<string> products = new List<string>

        {

            "iPhone 14",

            "iPhone 15",

            "iPad Pro",

            "MacBook Air",

            "MacBook Pro",

            "Apple Watch",

            "Samsung Galaxy",

            "Sony Headphones",

            "iMac",

            "iPod Nano"

        };

        Console.Write("Search Product: ");

        string input = Console.ReadLine();

        Console.WriteLine("\nSearch Results:");

        bool found = false;

        foreach (string product in products)

        {

            if (product.ToLower().Contains(input.ToLower()))

            {

                Console.WriteLine(product);

                found = true;

            }

        }

        if (!found)

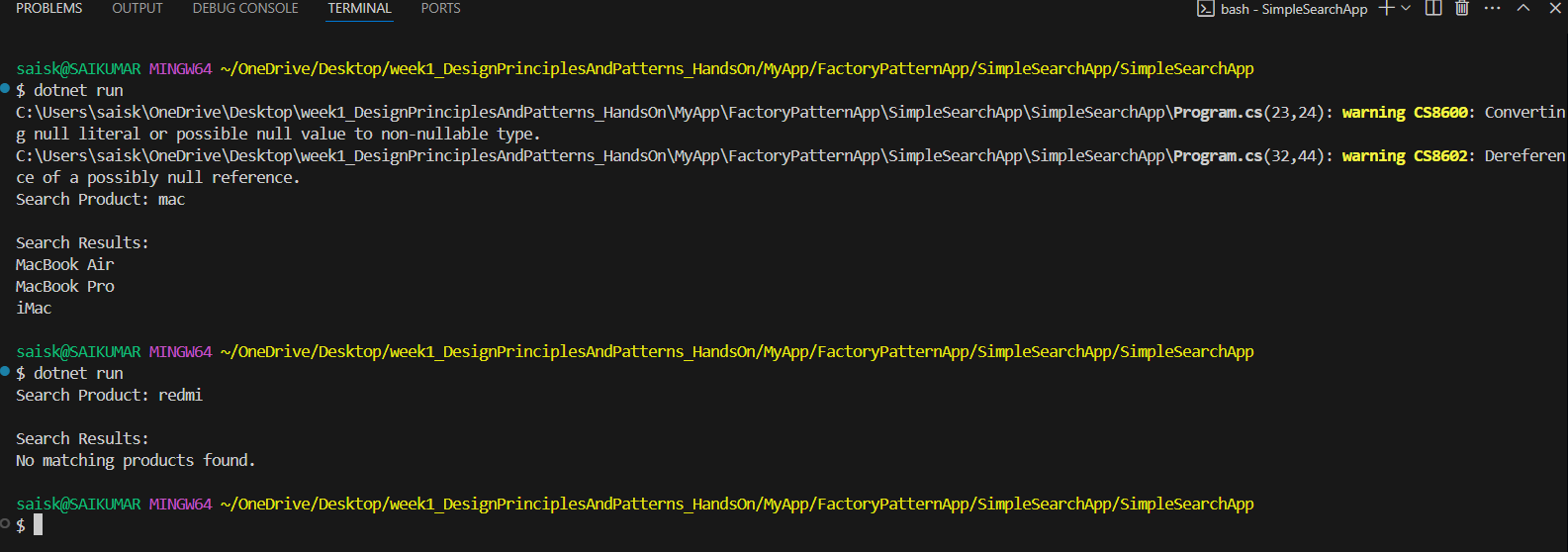
        {

            Console.WriteLine("No matching products found.");

        }

    }

}

**Output**

**Exercise 7: Financial Forecasting**

**Code:**

using System;

class Program

{

    static void Main()

    {

        Console.Write("Enter your income: ₹ ");

        decimal income = Convert.ToDecimal(Console.ReadLine());

        Console.Write("Enter your expenses: ₹ ");

        decimal expenses = Convert.ToDecimal(Console.ReadLine());

        decimal savings = income - expenses;

        Console.WriteLine($"\nYour savings this month: ₹ {savings}");

        if (savings > 0)

            Console.WriteLine("Good job! You're saving money.");

        else if (savings < 0)

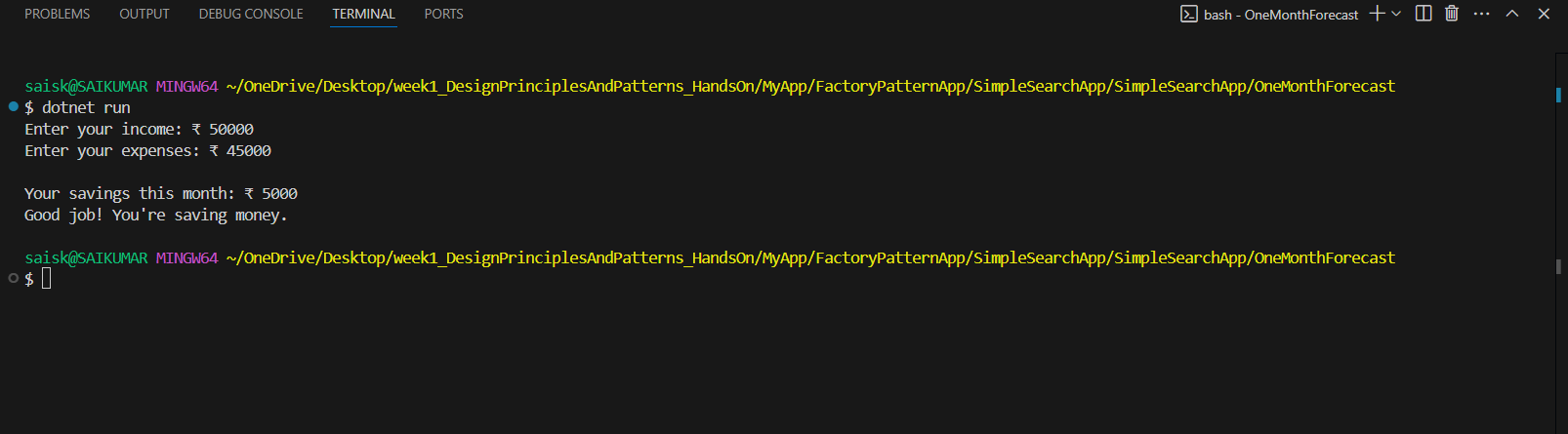
            Console.WriteLine("Warning! You're spending more than you earn.");

        else

            Console.WriteLine("You broke even this month.");

    }

}

**Output:**