**Exercise 3: Implementing the Builder Pattern**

**Scenario:**

You are developing a system to create complex objects such as a Computer with multiple optional parts. Use the Builder Pattern to manage the construction process.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **BuilderPatternExample**.
2. **Define a Product Class:**
   * Create a class **Computer** with attributes like **CPU**, **RAM**, **Storage**, etc.
3. **Implement the Builder Class:**
   * Create a static nested Builder class inside Computer with methods to set each attribute.
   * Provide a **build()** method in the Builder class that returns an instance of Computer.
4. **Implement the Builder Pattern:**
   * Ensure that the **Computer** class has a private constructor that takes the **Builder** as a parameter.
5. **Test the Builder Implementation:**
   * Create a test class to demonstrate the creation of different configurations of Computer using the Builder pattern.

**Solution:**  
  
using System;

public class Computer

{

public string CPU { get; }

public string RAM { get; }

public string Storage { get; }

public string GPU { get; }

public string Motherboard { get; }

private Computer(Builder builder)

{

CPU = builder.CPU;

RAM = builder.RAM;

Storage = builder.Storage;

GPU = builder.GPU;

Motherboard = builder.Motherboard;

}

public class Builder

{

public string CPU { get; private set; }

public string RAM { get; private set; }

public string Storage { get; private set; }

public string GPU { get; private set; }

public string Motherboard { get; private set; }

public Builder SetCPU(string cpu)

{

CPU = cpu;

return this;

}

public Builder SetRAM(string ram)

{

RAM = ram;

return this;

}

public Builder SetStorage(string storage)

{

Storage = storage;

return this;

}

public Builder SetGPU(string gpu)

{

GPU = gpu;

return this;

}

public Builder SetMotherboard(string motherboard)

{

Motherboard = motherboard;

return this;

}

public Computer Build()

{

return new Computer(this);

}

}

public void Display()

{

Console.WriteLine($"CPU: {CPU}");

Console.WriteLine($"RAM: {RAM}");

Console.WriteLine($"Storage: {Storage}");

Console.WriteLine($"GPU: {GPU}");

Console.WriteLine($"Motherboard: {Motherboard}");

Console.WriteLine();

}

}

public class Program

{

public static void Main()

{

var gamingPc = new Computer.Builder()

.SetCPU("Intel i9")

.SetRAM("32GB")

.SetStorage("1TB SSD")

.SetGPU("NVIDIA RTX 4080")

.SetMotherboard("ASUS ROG")

.Build();

var officePc = new Computer.Builder()

.SetCPU("Intel i5")

.SetRAM("16GB")

.SetStorage("512GB SSD")

.Build();

gamingPc.Display();

officePc.Display();

}

}

