Exercise 3: Implementing the Builder Pattern

# 🎯 Objective

To develop a system for creating complex objects such as a Computer with multiple optional parts using the Builder Pattern.

# ✅ Java Code – BuilderPatternExample.java

// Step 2 & 3: Product Class with Nested Builder Class  
class Computer {  
 private String CPU;  
 private String RAM;  
 private String storage;  
 private String graphicsCard;  
  
 private Computer(Builder builder) {  
 this.CPU = builder.CPU;  
 this.RAM = builder.RAM;  
 this.storage = builder.storage;  
 this.graphicsCard = builder.graphicsCard;  
 }  
  
 public String getCPU() { return CPU; }  
 public String getRAM() { return RAM; }  
 public String getStorage() { return storage; }  
 public String getGraphicsCard() { return graphicsCard; }  
  
 public static class Builder {  
 private String CPU;  
 private String RAM;  
 private String storage;  
 private String graphicsCard;  
  
 public Builder setCPU(String CPU) {  
 this.CPU = CPU;  
 return this;  
 }  
  
 public Builder setRAM(String RAM) {  
 this.RAM = RAM;  
 return this;  
 }  
  
 public Builder setStorage(String storage) {  
 this.storage = storage;  
 return this;  
 }  
  
 public Builder setGraphicsCard(String graphicsCard) {  
 this.graphicsCard = graphicsCard;  
 return this;  
 }  
  
 public Computer build() {  
 return new Computer(this);  
 }  
 }  
  
 @Override  
 public String toString() {  
 return "Computer [CPU=" + CPU + ", RAM=" + RAM + ", Storage=" + storage +   
 ", GraphicsCard=" + graphicsCard + "]";  
 }  
}  
  
// Step 6: Test Class  
public class BuilderPatternExample {  
 public static void main(String[] args) {  
 Computer basicComputer = new Computer.Builder()  
 .setCPU("Intel i3")  
 .setRAM("8GB")  
 .build();  
  
 Computer gamingComputer = new Computer.Builder()  
 .setCPU("AMD Ryzen 7")  
 .setRAM("32GB")  
 .setStorage("1TB SSD")  
 .setGraphicsCard("NVIDIA RTX 4070")  
 .build();  
  
 System.out.println("Basic Computer: " + basicComputer);  
 System.out.println("Gaming Computer: " + gamingComputer);  
 }  
}

# ✅ Sample Output

Basic Computer: Computer [CPU=Intel i3, RAM=8GB, Storage=null, GraphicsCard=null]  
Gaming Computer: Computer [CPU=AMD Ryzen 7, RAM=32GB, Storage=1TB SSD, GraphicsCard=NVIDIA RTX 4070]

# ✅ Key Benefits of Builder Pattern

|  |  |
| --- | --- |
| Feature | Description |
| Simplifies Construction | Builds complex objects step-by-step |
| Improves Readability | Clear and readable object creation |
| Flexible & Scalable | Easy to add new optional parameters |
| Immutable Product | Once built, the object is in a consistent, final state |