**Exercise 3: Implementing the builder pattern**

**Computer.java**

public class Computer {

private String cpu;

private String ram;

private String storage;

private String graphicsCard;

private String operatingSystem;

private Computer(Builder builder) {

this.cpu = builder.cpu;

this.ram = builder.ram;

this.storage = builder.storage;

this.graphicsCard = builder.graphicsCard;

this.operatingSystem = builder.operatingSystem;

}

public static class Builder {

private String cpu;

private String ram;

private String storage;

private String graphicsCard;

private String operatingSystem;

public Builder(String cpu, String ram) {

this.cpu = cpu;

this.ram = ram;

}

public Builder setStorage(String storage) {

this.storage = storage;

return this;

}

public Builder setGraphicsCard(String graphicsCard) {

this.graphicsCard = graphicsCard;

return this;

}

public Builder setOperatingSystem(String os) {

this.operatingSystem = os;

return this;

}

public Computer build() {

return new Computer(this);

}

}

public void showSpecs() {

System.***out***.println("CPU: " + cpu);

System.***out***.println("RAM: " + ram);

System.***out***.println("Storage: " + (storage != null ? storage : "Not included"));

System.***out***.println("Graphics Card: " + (graphicsCard != null ? graphicsCard : "Not included"));

System.***out***.println("Operating System: " + (operatingSystem != null ? operatingSystem : "Not included"));

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Computer basicComputer = new Computer.Builder("Intel i3", "4GB").build();

System.***out***.println("Basic Computer Configuration:");

basicComputer.showSpecs();

System.***out***.println("\n----------------------------------\n");

Computer gamingComputer = new Computer.Builder("Intel i9", "32GB")

.setStorage("1TB SSD")

.setGraphicsCard("NVIDIA RTX 4080")

.setOperatingSystem("Windows 11 Pro")

.build();

System.***out***.println("Gaming Computer Configuration:");

gamingComputer.showSpecs();

}

}

Output:

