**Design Patterns and Principles – Exercise 3**

**Builder Pattern Example**

**Main.java**

public class Main {

public static void main(String[] args) {

Computer basicComputer = new Computer.Builder()

.setCPU("Intel i5")

.setRAM("8GB")

.setStorage("256GB SSD")

.build();

basicComputer.showSpecs();

Computer gamingComputer = new Computer.Builder()

.setCPU("Intel i9")

.setRAM("32GB")

.setStorage("1TB SSD")

.setGPU("NVIDIA RTX 4080")

.build();

gamingComputer.showSpecs();

Computer budgetComputer = new Computer.Builder()

.setCPU("AMD Ryzen 3")

.setRAM("4GB")

.build();

budgetComputer.showSpecs();

}

}

**Computer.java**

public class Computer {

private String CPU;

private String RAM;

private String storage;

private String GPU;

private Computer(Builder builder) {

this.CPU = builder.CPU;

this.RAM = builder.RAM;

this.storage = builder.storage;

this.GPU = builder.GPU;

}

public static class Builder {

private String CPU;

private String RAM;

private String storage;

private String GPU;

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 setGPU(String GPU) {

this.GPU = GPU;

return this;

}

public Computer build() {

return new Computer(this);

}

}

public void showSpecs() {

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

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

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

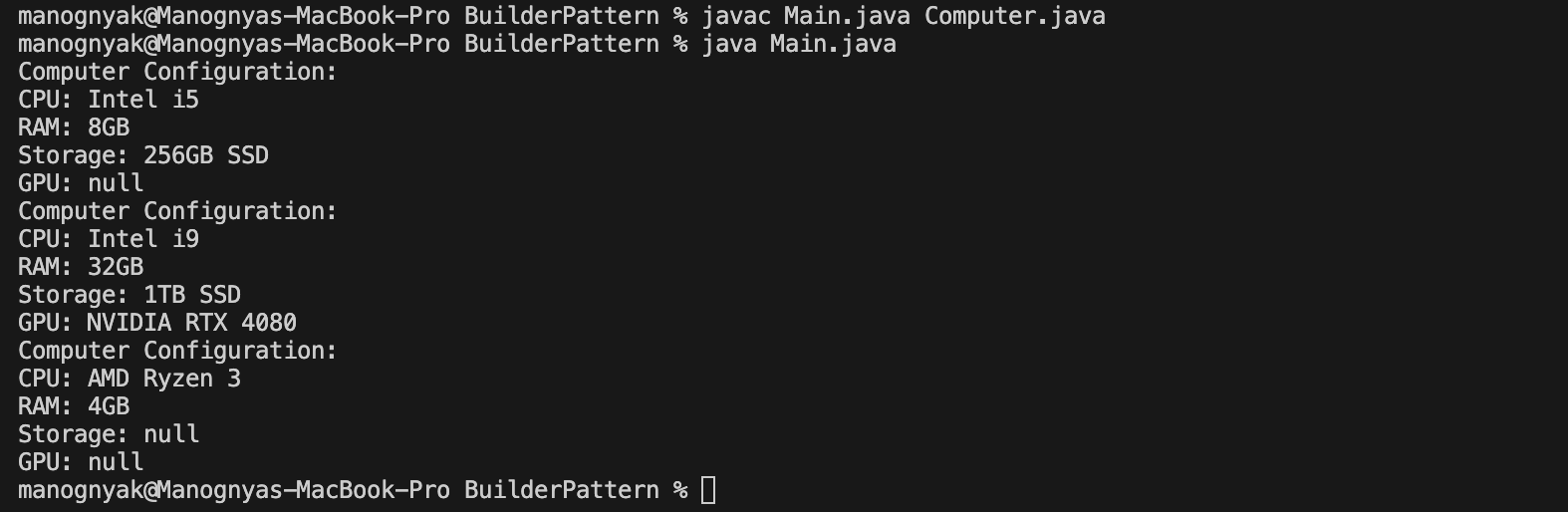
System.out.println("Storage: " + storage);

System.out.println("GPU: " + GPU);

}

}

**Output**

****