WEEK 1

Additional Hands-On

**SOLID PRINCIPLES AND DESIGN PATTERNS:  
EX:3 Implementing the Builder Pattern**

The reason to use Builder design pattern for the desired construction/ building of a computer is that,

* It allows step-by-step object construction.
* It improves code readability.
* It supports immutability by constructing the object only once, at the end.
* It avoids telescoping constructors.

**Product and Builder class (Definition and Implementation):**

public 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 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"));

}

public static class Builder {

private String CPU;

private String RAM;

private String storage;

private String graphicsCard;

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 Computer build() {

return new Computer(this);

}

}

}

* + - The Builder class is a static nested class.
    - The build() method constructs the final Computer object by calling the private constructor of Computer.

**Testing the builder class using Main class:**

public class Main {

public static void main(String[] args) {

Computer basicComputer = new Computer.Builder("Intel i5", "8GB").build();

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

basicComputer.showSpecs();

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

.setStorage("1TB SSD")

.setGraphicsCard("NVIDIA RTX 3080")

.build();

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

gamingComputer.showSpecs();

}

}

**** **TESTING OUTPUT:**