**Design Pattern And Principles**

**EXERCISE 3: Implementing the Building Pattern**

**Source Code**

// Main.java

class Computer {

// Required components

private final String cpu;

private final String ram;

// Optional components

private final String storage;

private final String graphicsCard;

private final String operatingSystem;

// Private constructor

private Computer(Builder builder) {

this.cpu = builder.cpu;

this.ram = builder.ram;

this.storage = builder.storage;

this.graphicsCard = builder.graphicsCard;

this.operatingSystem = builder.operatingSystem;

}

// Static nested Builder class

public static class Builder {

// Required components

private final String cpu;

private final String ram;

// Optional components

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 operatingSystem) {

this.operatingSystem = operatingSystem;

return this;

}

public Computer build() {

return new Computer(this);

}

}

public void displayConfiguration() {

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

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

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

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

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

System.out.println();

}

}

public class Main {

public static void main(String[] args) {

Computer officePC = new Computer.Builder("Intel i3", "8GB").build();

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

.setStorage("1TB SSD")

.setGraphicsCard("NVIDIA RTX 4090")

.setOperatingSystem("Windows 11 Pro")

.build();

System.out.println("=== Office PC Configuration ===");

officePC.displayConfiguration();

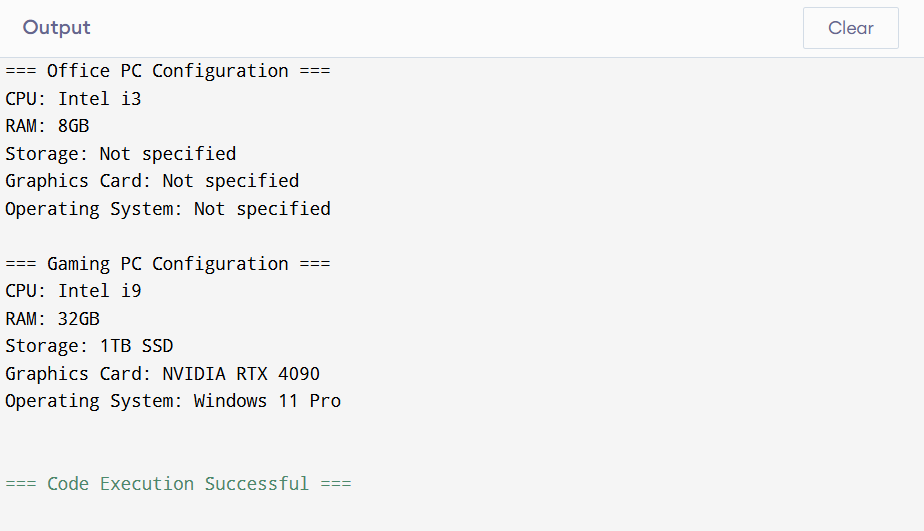
System.out.println("=== Gaming PC Configuration ===");

gamingPC.displayConfiguration();

}

}

**OUTPUT**

****