**Module 1**

**Exercise 3: Implementing the Builder Pattern**

**Program:**

public class Main {

public static void main(String[] args) {

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

.build();

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

.setStorage("1TB SSD")

.setGraphicsCard("NVIDIA RTX 4070")

.setOperatingSystem("Windows 11 Pro")

.build();

Computer devMachine = new Computer.Builder("AMD Ryzen 7", "16GB")

.setStorage("512GB SSD")

.setOperatingSystem("Ubuntu 22.04")

.build();

System.out.println("\n--- Basic Computer ---");

basicComputer.showConfig();

System.out.println("\n--- Gaming Computer ---");

gamingComputer.showConfig();

System.out.println("\n--- Developer Machine ---");

devMachine.showConfig();

}

public static 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 operatingSystem) {

this.operatingSystem = operatingSystem;

return this;

}

public Computer build() {

return new Computer(this);

}

}

public void showConfig() {

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 installed"));

}

}

}

**OUTPUT:**

