**MODULE-1**

**DESIGN PATTERN AND PRINCIPLE**

SUPERSET ID:6407550

**Exercise 3: Implementing the Builder Pattern**

**Main.java**

package builder;

public class Main {

    public static void main(String[] args) {

        // Basic configuration

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

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

        System.out.println(basicComputer);

        // High-end configuration

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

                                    .setStorage("1TB SSD")

                                    .setGraphicsCard("NVIDIA RTX 4080")

                                    .setOperatingSystem("Windows 11 Pro")

                                    .build();

        System.out.println("\nGaming Computer:");

        System.out.println(gamingComputer);

        // Mid-range configuration

        Computer workComputer = new Computer.Builder("AMD Ryzen 5", "16GB")

                                 .setStorage("512GB SSD")

                                 .setOperatingSystem("Ubuntu Linux")

                                 .build();

        System.out.println("\nWork Computer:");

        System.out.println(workComputer);

    }

}

**Computer.java**

package builder;

public 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 {

        private final String cpu;

        private final 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);

        }

    }

    @Override

    public String toString() {

        return "Computer Config:\n" +

               "CPU: " + cpu + "\n" +

               "RAM: " + ram + "\n" +

               "Storage: " + (storage != null ? storage : "Not Included") + "\n" +

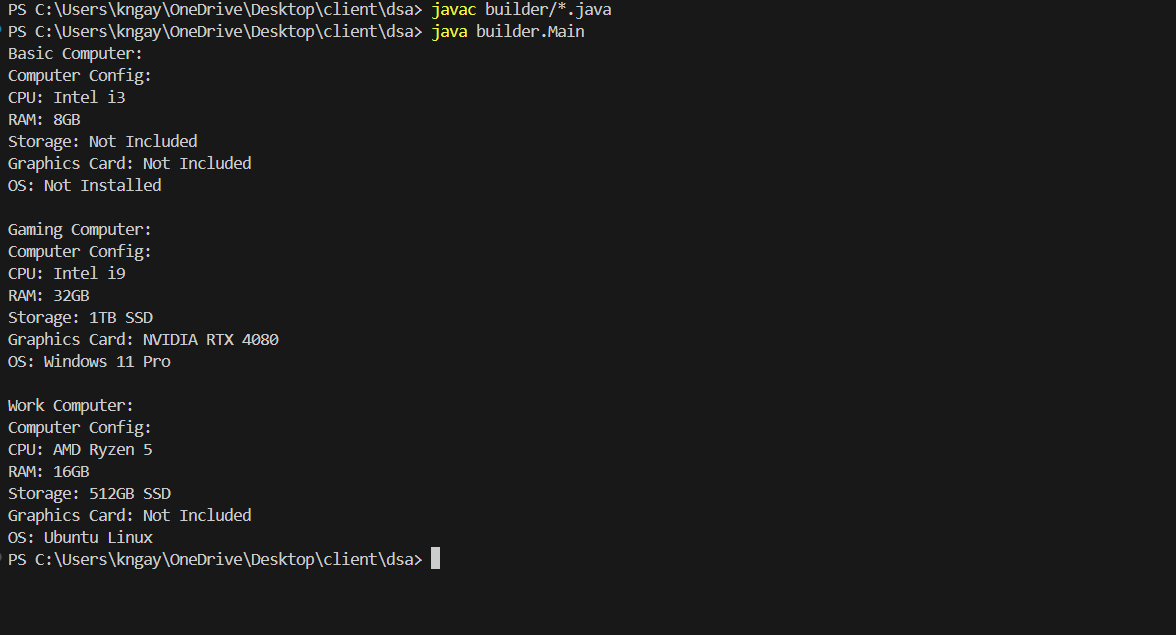
               "Graphics Card: " + (graphicsCard != null ? graphicsCard : "Not Included") + "\n" +

               "OS: " + (operatingSystem != null ? operatingSystem : "Not Installed");

    }

}

**OUTPUT:**

****