**Design Patterns and Principles – Exercise 3**

**Builder Pattern Example**

**Main.java**

public class Main {

    public static void main(String[] args){

        Computer basic = new Computer.Builder()

            .setCPU("Intel i5")

            .setRAM("8GB")

            .setStorage("500GB HDD")

            .build();

        Computer gaming = new Computer.Builder()

            .setCPU("Intel i9")

            .setRAM("32GB")

            .setStorage("1TB SSD")

            .build();

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

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

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

        System.out.println("Storage: " + basic.getStorage());

        System.out.println();

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

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

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

        System.out.println("Storage: " + gaming.getStorage());

    }

}

**Computer.java**

public class Computer{

    private String CPU;

    private String RAM;

    private String storage;

    //Private Constructor

    private Computer(Builder builder){

        this.CPU = builder.CPU;

        this.RAM = builder.RAM;

        this.storage = builder.storage;

    }

    //Getters for the attributes

    public String getCPU(){

        return CPU;

    }

    public String getRAM(){

        return RAM;

    }

    public String getStorage(){

        return storage;

    }

    // Static Nested Builder class

    public static class Builder{

        private String CPU;

        private String RAM;

        private String storage;

        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 Computer build(){

            return new Computer(this);

        }

    }

}

**Output**

