**Exercise 3: Implementing the Builder Pattern**

**Program:**

Computer.java

package DesignPatterns.BuilderPatternExample;

public class Computer {

    private final String cpu;

    private final String RAM;

    private final String Storage;

    Computer(Builder builder){

        this.cpu = builder.cpu;

        this.RAM = builder.RAM;

        this.Storage = builder.Storage;

    }

    public String getCpu() {

        return cpu;

    }

    public String getRAM() {

         return RAM;

         }

    public String getStorage() {

         return Storage;

        }

    public static class Builder{

        private final String cpu;

        private String RAM;

        private String Storage;

        public Builder(String cpu){

            this.cpu = cpu;

        }

        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);

        }

    }

}

Main.java

public class Main {

    public static void main(String[] args) {

        Computer computer1 = new Computer.Builder("Intel").setRAM("16GB").setStorage("512SSD").Build();

        System.out.println(computer1.getCpu());

        System.out.println(computer1.getRAM());

        System.out.println(computer1.getStorage());

    }

}

A black screen with white text

AI-generated content may be incorrect.