**Exercise 3: Implementing the Builder Pattern**

**Scenario:**

You are developing a system to create complex objects such as a Computer with multiple optional parts. Use the Builder Pattern to manage the construction process.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **BuilderPatternExample**.
2. **Define a Product Class:**
   * Create a class **Computer** with attributes like **CPU**, **RAM**, **Storage**, etc.
3. **Implement the Builder Class:**
   * Create a static nested Builder class inside Computer with methods to set each attribute.
   * Provide a **build()** method in the Builder class that returns an instance of Computer.
4. **Implement the Builder Pattern:**
   * Ensure that the **Computer** class has a private constructor that takes the **Builder** as a parameter.
5. **Test the Builder Implementation:**
   * Create a test class to demonstrate the creation of different configurations of Computer using the Builder pattern.

## Answer:

Computer.java

public class Computer {

    String CPU;

    String RAM;

    String storage;

    private Computer(ComputerBuilder 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;

    }

    @Override

    public String toString() {

        return "\nCPU = " + CPU + '\n' + "RAM = " + RAM + '\n' +"Storage = " + storage+'\n';

    }

    public static class ComputerBuilder {

        private String CPU;

        private String RAM;

        private String storage = "256GB SSD";

        public ComputerBuilder(String cpu, String ram) {

            this.CPU = cpu;

            this.RAM = ram;

        }

        public ComputerBuilder storage(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 basic = new Computer.ComputerBuilder("Intel i3", "8GB").build();

        System.out.println("Basic: " + basic);

        Computer work = new Computer.ComputerBuilder("Intel i5", "16GB").storage("512GB SSD").build();

        System.out.println("Work: " + work);

        Computer gaming = new Computer.ComputerBuilder("AMD Ryzen 7", "16GB") .storage("1TB NVMe SSD").build();

        System.out.println("Gaming: " + gaming);

    }

}

Output:

