supply pre-written solution

**ChatGPT said:**

Here’s the full **pre-written solution** for the **Computer Builder** exercise, following all your specs:

**✅ Computer.java**

java

CopyEdit

public class Computer {

// Required

private final String cpu;

private final String ram;

private final String storage;

// Optional

private final String gpu;

private final String cooling;

private final String powerSupply;

private Computer(Builder builder) {

this.cpu = builder.cpu;

this.ram = builder.ram;

this.storage = builder.storage;

this.gpu = builder.gpu;

this.cooling = builder.cooling;

this.powerSupply = builder.powerSupply;

}

public static abstract class Builder {

// Required

protected String cpu;

protected String ram;

protected String storage;

// Optional

protected String gpu;

protected String cooling;

protected String powerSupply;

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 Builder setGpu(String gpu) {

this.gpu = gpu;

return this;

}

public Builder setCooling(String cooling) {

this.cooling = cooling;

return this;

}

public Builder setPowerSupply(String powerSupply) {

this.powerSupply = powerSupply;

return this;

}

public abstract Computer build();

}

@Override

public String toString() {

return String.format("CPU: %s, RAM: %s, Storage: %s, GPU: %s, Cooling: %s, PSU: %s",

cpu, ram, storage,

gpu != null ? gpu : "None",

cooling != null ? cooling : "None",

powerSupply != null ? powerSupply : "None");

}

}

**✅ GamingPCBuilder.java**

java

CopyEdit

public class GamingPCBuilder extends Computer.Builder {

@Override

public Computer build() {

// Enforce required fields

if (cpu == null || ram == null || storage == null) {

throw new IllegalStateException("Missing required fields for Gaming PC");

}

return new Computer(this);

}

}

**✅ OfficePCBuilder.java**

java

CopyEdit

public class OfficePCBuilder extends Computer.Builder {

@Override

public Computer build() {

if (cpu == null || ram == null || storage == null) {

throw new IllegalStateException("Missing required fields for Office PC");

}

return new Computer(this);

}

}

**✅ WorkstationPCBuilder.java**

java

CopyEdit

public class WorkstationPCBuilder extends Computer.Builder {

@Override

public Computer build() {

if (cpu == null || ram == null || storage == null) {

throw new IllegalStateException("Missing required fields for Workstation");

}

return new Computer(this);

}

}

**✅ Director.java**

java

CopyEdit

public class Director {

public Computer buildGamingPC() {

return new GamingPCBuilder()

.setCpu("Intel i9")

.setRam("32GB")

.setStorage("1TB SSD")

.setGpu("NVIDIA RTX 4090")

.setCooling("Water Cooling")

.setPowerSupply("850W")

.build();

}

public Computer buildOfficePC() {

return new OfficePCBuilder()

.setCpu("Intel i5")

.setRam("16GB")

.setStorage("512GB SSD")

.setCooling("Air Cooling")

.setPowerSupply("500W")

.build();

}

public Computer buildWorkstation() {

return new WorkstationPCBuilder()

.setCpu("AMD Ryzen 9")

.setRam("64GB")

.setStorage("2TB NVMe")

.setGpu("NVIDIA RTX 3090")

.setCooling("Air Cooling")

.setPowerSupply("1000W")

.build();

}

}

**✅ Main.java**

java

CopyEdit

public class Main {

public static void main(String[] args) {

Director director = new Director();

Computer gamingPC = director.buildGamingPC();

Computer officePC = director.buildOfficePC();

Computer workstation = director.buildWorkstation();

System.out.println("Gaming PC Configuration:");

System.out.println(gamingPC);

System.out.println();

System.out.println("Office PC Configuration:");

System.out.println(officePC);

System.out.println();

System.out.println("Workstation Configuration:");

System.out.println(workstation);

}

}

**🔍 Notes:**

* The Builder class is **abstract** so it can be extended by specialized builders.
* Director encapsulates common setups so the client doesn't have to remember all config details.
* Optional fields (gpu, cooling, powerSupply) are easily excluded without modifying constructors or creating subclasses.
* Validation in each builder ensures required fields are present before building.