**Implementing the Builder Pattern**

Computer.java  
  
public class Computer {

private String cpu;

private String ram;

private String storage;

private Computer(Builder builder){

this.cpu = builder.cpu;

this.ram = builder.ram;

this.storage = builder.storage;

}

public static class Builder{

private String cpu;

private String ram;

private String storage;

public Builder(String cpu, String ram) {

this.cpu = cpu;

this.ram = ram;

}

public Builder setStorage(String storage) {

this.storage = storage;

return this;

}

public Computer build() {

return new Computer(this);

}

}

@Override

public String toString(){

return "Computer [CPU=" + cpu + ", RAM=" + ram + ", Storage=" + storage + "]";

}

}

TestBuilderPattern.java  
  
// Source code is decompiled from a .class file using FernFlower decompiler.

public class Computer {

private String cpu;

private String ram;

private String storage;

private Computer(Computer$Builder var1) {

this.cpu = var1.cpu;

this.ram = var1.ram;

this.storage = var1.storage;

}

public String toString() {

return "Computer [CPU=" + this.cpu + ", RAM=" + this.ram + ", Storage=" + this.storage + "]";

}

}

Output

