**Exercise 3: Implementing the Builder Pattern**

**Code:**

**Computer.java Class: -**

public class Computer {  
 public String CPU;  
 public String RAM;  
 public String Storage;  
  
 private Computer(Builder builder){  
 this.CPU=builder.CPU;  
 this.RAM=builder.RAM;  
 this.Storage=builder.Storage;  
 }  
  
 public static class Builder{  
 public String CPU;  
 public String RAM;  
 public 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);  
 }  
 }  
 @Override  
 public String toString(){  
 return "Computer has "+CPU+" CPU, "+RAM+" RAM, "+Storage+" storage.";  
 }  
}

**TestClass.java (Main class): -**

public class TestClass {  
 public static void main(String[] args){  
 Computer comp=new Computer.Builder()  
 .setCPU("Intel i5")  
 .setRAM("32GB")  
 .setStorage("1TB SSD")  
 .build();  
 System.*out*.println(comp.toString());  
 Computer comp2=new Computer.Builder()  
 .setRAM("16GB")  
 .setCPU("Intel i9")  
 .build();  
 System.*out*.println(comp2.toString());  
 }  
}

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**