The Single Responsibility Principle states that **a class should do one thing and therefore it should have only a single reason to change**.

Let’s take a scenario

We have a class called **Book.java**

Graphical user interface, text, application

Description automatically generated

and a class called **Invoice.java**

**Graphical user interface, text, application

Description automatically generated**

If you’ll see the invoice class closely that you’ll find that there is two methods called **calculateTotal()** and **printInvoice()**

**Here we have two reason two changes :**

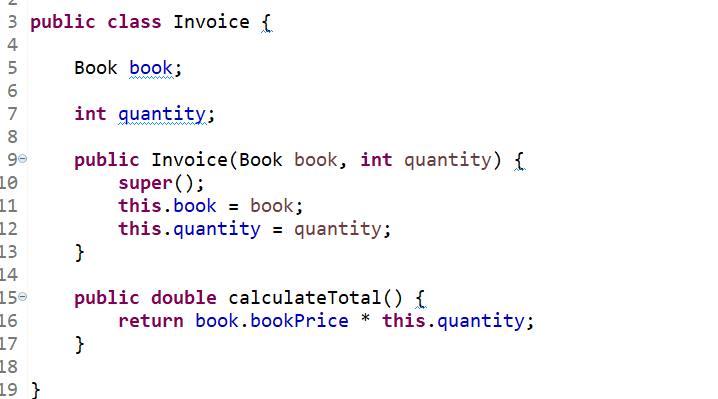
1. In future we can have different printing logic for print invoice, it will lead to change in printInvoice() method.
2. For now we don’t consider taxes applied on book but it is obvious that in future we have to add taxes on book too so in here the calculation logic will also change.

So here, this class has two reason two. That’s why this class is breaking SRP.

**Solution :**

According to SRP, we should wrap up both the functionality into separate classes.

From Invoice.java we have separated printing logic as you can see below :



Here we have created a new class InvoicePrinter.java for printing logic of our application.

Graphical user interface, text, application

Description automatically generated

And here is our main class :

Text

Description automatically generated

Now our application is following the SRP.