Project 2: Bookstore Order Calculator

Due: Monday, September 14 at 11:59 PM

Description: You've decided to start your own online bookstore! Unlike other online retailers, which overwhelm their customers with tons of mediocre products, you will only sell your three favorite books. To automate customer purchases, you need to write a program that calculates the total price of an order.

Objectives: Your program will be graded according to the rubric below. Please review each objective before submitting your work so you don't lose easy points. For example, objective 2 is not intended to be difficult, but there were a handful of careless students who missed these points last semester.

1. Create a new project and class in Eclipse. (5 points)
2. Add a comment with your name(s) on the first line of your code. (5 points)
3. Add the main method to the class. (5 points)
4. Use variables to store three book titles and their authors. (10 points)
5. Use variables to store the price of each book. (10 points)
6. Use variables to store the number of copies purchased of each book. (10 points)
7. Use a constant variable to store the tax rate (8.75%). (5 points)
8. Calculate the pretax total, sales tax, and total order cost using arithmetic operators and the variables described above. Store the calculation results in variables. (20 points)
9. Print the order information to the console. When printing dollar amounts, round each value to two decimal places (i.e., to the nearest penny). (20 points)
10. Use meaningful variable names, consistent indentation, and whitespace (blank lines and spaces) to make your code readable. Add comments where appropriate to explain the overall steps of your program. (10 points)

Note for CS 1323 Students: You are not required to have a TA check your work this week, although you are welcome to attend office hours for this purpose. Please do not attend the CS 1324 lab sections.

Note for CS 1324 Students: Starting this week, projects will be completed with a partner. This is known as "pair programming." You and your partner will write the program together on a single laptop. One person acts as the driver, who controls the mouse and keyboard and inputs the code. The other person acts as the navigator, who makes suggestions and asks questions as the driver types. The navigator also supports the driver by looking up answers to questions in the textbook or the API documentation on a second laptop. Every half hour, you and your partner will switch roles.

If you do not complete the project during lab, finish writing the program individually during the week and submit before the deadline. Make sure you and your partner both have a copy of your code before leaving. Whether you finish in lab or not, you both must submit the project to Canvas.

Sample Output: Below is an example of the information that the program should print to the console:

```
Purchase 2 copies of Three Sisters, Bi Feiyu. Each copy costs $18.99.
Purchase 4 copies of Song of Solomon, Toni Morrison. Each copy costs
$11.99.
Purchase 1 copy of Owls Don't Have to Mean Death, Chip Livingston. Each
copy costs $80.50.
Total before tax: $166.44
Sales tax: $14.56
The total cost of your order will be $181.00
```

Your output should have the same format, but the book titles, authors, price per copy, and number of copies may be different.

Create a New Project and Class: Refer back to the Project 1 handout for instructions on creating a new project and class in Eclipse. When creating the project, make sure you don't create the module-info.java file. Name your class `Project2`.

Add the Main Method: After the class is created, you need to add the main method. When you finish, your source file should contain the following code, but with your name(s) appearing in the class name:

```
public class Project2
{
	public static void main(String[] args)
	{
		// The remaining code goes here.
	}
}
```

Write the rest of the program between the curly braces of the main method. This is the space marked with the comment "The remaining code goes here." (Note that this comment does not need to be included in your program.)

Store Data in Variables: You get to choose the books for sale in your bookstore and the price per copy of each book. Use variables to store this information in the program. Each variable needs to have the appropriate type for the data that it stores as well as a descriptive, legal identifier. (Please use reasonable values for the prices. Too low or too high and your store will go out of business!)

No User Input: Customers choose the number of copies of each book that they want to purchase. In future projects, we will use a Scanner to get data like this from the user. Since we don't have enough experience with Scanner objects, however, we will hard code these values instead. This means that you get to choose the number of copies being ordered. As with the book titles and prices, use variables to store this information in the program. (Assume the user orders at least one copy of each book.)

Tax Rate: The sales tax rate in Norman is 8.75%. This value is unlikely to change very often, so we should store it in a constant variable. Recall that a variable can be made constant by using the `final` keyword. By convention, constant variables are given names with all capital letters and underscores between words.

Calculate the Pretax Total, Sales Tax, and Total Order Cost: Using the variables described above, calculate the cost of the order before tax, the sales tax, and the total cost with tax. Store the result of each calculation in a variable. If done correctly, the price or number of copies of any book can be changed (by changing the value stored in the corresponding variable), and the calculations will automatically be updated to reflect the change. This is the advantage of using variables in your calculations instead of literal values.

Print Order Information to the Console: The last thing your program should do is output information about the order to the console. Format the output so that it matches the sample output above. As you saw in Project 1, information can be printed to the console using the method `System.out.println`. Below is an example that prints the sales tax:

```
System.out.println("Sales tax: $" + salesTax);
```

Note that the plus operator converts the value stored in `salesTax` to a String and concatenates (i.e., connects) it to the end of `"Sales tax: $"`.

Round to the Nearest Penny: There is a problem with the code given in the last paragraph. If the value stored in `salesTax` is a double, then the value printed to the console will include a fraction of a penny. We want to print the value rounded to two decimal places. An easy way to do this is to use the `format` method of the String class. Below is the modified code:

```
System.out.println("Sales tax: $" + String.format("%.2f", salesTax));
```

The variable `salesTax` is the second argument in the method. The first argument, `"%.2f"`, is a format string that tells the method how many digits to keep after the decimal place. Use this method to print the pretax total, sales tax, and total order cost rounded to the nearest penny.

Upload Code to Zybooks: After you complete each project, you need to upload your source code to Zybooks so it can be graded. If you created the folder structure described earlier, your code is in the folder “Intro to Programming\Projects\Project 2\src”. The code is contained in the file Project2.java. Note that .java files are simply text files that contain Java code. They can be opened with any text editor (e.g., Notepad or TextEdit).

Upload your .java file to Zybooks on the Project 2 assignment page. This requires a few steps. Click the “Submit Assignment” button in the top-right corner. This will reveal a button near the bottom of the page with the text “Choose File.” Click this button and browse to the location of your .java file. Finally, click the “Submit for grading” button below the “Choose on hard drive” button.

If Zybooks won't accept the file, make sure you're submitting a .java file, not a .class file. Make sure to see if the output you obtained and the output which I gave match to obtain complete grade.