

# CIS 255 • Java Programming

## Project 1: File Input and Output

A local business serves customers in four cities. It stores information about orders made through its website in the file `webOrders.txt`, in which each order's information consists of two values: the zip code (a five-digit whole number) and the pre-tax total. The following is an example of how a typical order appears in the file:

```
31591 39.95
```

The business wants to separate these orders into separate text files, one for each city. The information about each order should be written on a separate line and should include the pre-tax total, the sales tax amount, and the total with tax (all written with commas as necessary and with two digits after the decimal point). Each city has a different sales tax rate according to the following table:

Zip Code	City Name	Filename	Sales Tax Rate
31591	Adamsdale	adamsdale.txt	7.00%
31594	Fairville	fairville.txt	10.00%
31596	Ironton	ironton.txt	9.00%
31597	Roly City	rolycity.txt	8.50%

Since the above example is in the zip code 31591, the program should write following line to the file `adamsdale.txt`:

```
39.95 2.80 42.75
```

Note that the zip code itself is not part of the file output (all of the orders in the same file would have the same zip code already). There should be some form of spacing between the three values on the same line (do not generate this output in sentence form). The number of orders in the original file is unknown, so the program must read all of the orders it encounters until it reaches the end of the file. If the input file cannot be opened, the program should display an error message and terminate. If an output file already exists, the program should overwrite it (it should not add to the existing contents of the file).

Once the program has completed processing the orders, it should display a message notifying the user that the output has been written to the files.

**POINTS: 75**

**BONUS (5 points):** There are only four valid zip codes for orders. If the zip code of an order read from the file does not match any of these values, the program should write both of the input values for that order on a single line in the file `cityError.txt`, with a space between the values and two digits after the decimal point in the pre-tax total.

**BONUS (5 points):** As the program writes the output to each file, it should calculate the sums of the pre-tax totals, the sales tax amounts, and the totals with tax (a separate set of totals for each file). After it has processed all of the orders in the input file, it should write these three totals (separated by spaces and with commas and two digits after the decimal point) as a final line of output in each file.

**BONUS (5 points):** The pre-tax total for each order will be less than \$100,000, the sales tax amount will be less than \$10,000, and the total with tax will be less than \$1,000,000. In the output file, the program should write these values in such a way that they line up in right-aligned columns.

**No bonus credit will be given unless the standard requirements of the assignment are met! Make sure that you have the basic functionality working regardless of whether or not you are able to complete the bonus portion successfully!**

Your code should include comments at the top listing your name, the course, the project number, and a brief description of the program. It should also include **at least five inline comments** in your code explaining what your code is doing at that given point.

The deadline is posted on Blackboard. **No late work will be accepted! The deadline will be strictly enforced by Blackboard!**

You must use the syntax and structures covered in the lectures and chapters covered thus far in the textbook unless you receive prior authorization from your instructor. **Any attempt to use syntax beyond the scope of the course may result in significantly reduced credit, even if your solution is fully functional.**

If you need assistance with your project code, e-mail your instructor **at least one day in advance** of the deadline. E-mail messages sent too close to the deadline may not receive a response in adequate time to make corrections to the project.

You may discuss general details of the project with your classmates, but **do not look at or borrow another student's project code!** In addition, **do not post the details of the project or ask for assistance with specifics of this project on any public discussion board, forum, or newsgroup!**