CS1321 Sales Tax Computation Tasks*

Jeffrey D. Oldham

2000 Feb 02

Contents

We propose to write an Internet-based sales tax system capable of calculating the appropriate sales tax on a group of taxable and tax-exempt items for any particular tax jurisdiction. Since this is a large project, we split the tasks into smaller pieces. Please choose one of the following tasks. Write your name next to your chosen task on the sign-up sheet on my office door HAS 201J. Sign-ups are filled on a first-come, first-served basis.

We describe each task and list the deadline for completion. For tasks requiring team work, the number of people in the team is listed. To ensure the project succeeds even if some task is not completed, each task will be completed by at least two teams.

Jeffrey D. Oldham may have already collected more information than he has listed here. It is probably worth talking with him for five minutes soon after starting work on your task.

1 Programming Tasks

1.1 What to Submit

Programmers should submit code, documentation how to use the code, and the test cases used to test the code. In other words, the software should be at the same level of a library that others should be able to use from just reading the documentation and that is as robust. All submissions should be in a computer-independent and operating system format, e.g., no Microsoft Word documents but text or HTML is acceptable. These will be posted on the class WWW site for all to peruse and use.

1.2 Parsing Input

task: Handle all input to the program, storing the information in the program's data structures. For example, parse the incoming addresses and lists of items, storing the information. Some of this code has already been written by Jeffrey.

number of teams: 2

team size: 1 person

deadline: 14 Mar

sources: Jeffrey's partial parsing code

related tasks: 1.8

1.3 Input Generation

task: Create addresses and item lists for use in testing the program. We should use some real-world addresses and some close-to-real-world item lists, but we can also create random data. Creating random addresses may be tricky since the USPS server will reject them, but we can just provide the ZIP+4 code in the address so the server is not necessary.

number of teams: 2

team size: 2 persons

deadline: 07 Mar

sources:

related tasks: 2.6, 1.4

1.4 Determining ZIP+4 Codes

task: Determine an address's ZIP+4 code using the USPS's server. Involves writing the information in the format required by the USPS ZIP+4 server, obtaining the answer, and parsing it. This requires some use of provided socket code. Also, determine what commercial products to do the same tasks are available and their costs.

number of teams: 2

team size: 2 persons

deadline: 14 Mar

sources: USPS WWW information, socket documentation, Jeffrey's socket code, WWW

related tasks: 1.2

1.5 Hash Tables

task: Write all code dealing with hash tables for tax rate and tax jurisdiction lookup. This involves reading data from files, hashing the data for use, and writing the code that converts an address to a tax rate and tax jurisdiction.

number of teams: 2

team size: 2 persons

deadline: 14 Mar

sources: hash table documentation, homework 3 code using hash tables

related tasks: 1.2

1.6 Tax Jurisdiction Coder

task: Investigate current regulations for rounding sales tax amounts for different real-world tax jurisdictions. Write the code for the tax jurisdiction class.

number of teams: 2

team size: 1 person

deadline: 14 Mar

sources: library, government documents, WWW, Jeffrey has a few notes

related tasks:

1.7 Integrator for Input and Tax Jurisdiction Portion

task: Collect together all the program code involving input to the program, the code to determine ZIP+4 postal codes, and the code to determine tax rates and jurisdictions, writing code to glue it together.

number of teams: 2

team size: 2 persons

deadline: 28 Mar

sources: code provided by the related tasks

related tasks: 1.2, 1.3, 1.4, 1.5

1.8 Taxability Lookup

task: Given an item and a tax jurisdiction, determine the item's taxability.

number of teams: 2

team size: 2 persons

deadline: 21 Mar

sources: Jeffrey's partial parsing code

related tasks: Since the taxability tables have similar format to the addresses and item lists, work closely with the Task 1.2 team.

1.9 Taxability Testing

task: Try to break the taxability lookup code (see Task 1.8). Write programs and random data to automatically try many different combinations to break the code.

number of teams: 2

team size: 1 person

deadline: 28 Mar

sources:

related tasks: 1.8

1.10 Master Integrator

task: Collect together all the program code, writing enough code to glue it together. This will involve the code provided by the Task 1.7 integrator, the code to determine taxability, and writing code to compute the answer.

number of teams: 2

team size: 3 persons

deadline: 11 Apr

sources: everyone else's code

related tasks: 1.7, 1.6, 1.8

1.11 Master Breaker

task: Try to break the final product. This involves both hands-on testing and writing code to automatically submit test cases (with known answers) and checking whether the answer is correct.

number of teams: 2

team size: 2 persons

deadline: 18 Apr

sources: code from Task 1.3 and 1.9

related tasks: 1.3, 1.9, 1.10

2 Nonprogramming Tasks

2.1 What to Submit

Investigators should submit eight to ten pages of text discussing the material, presenting background information, critiques, and recommendations. Be sure to include citations and a bibliography showing your sources. Please use sources that are among the most respected possible. For example, *The New York Times* is acceptable, but the *San Antonio Express-News* is not.

2.2 Investigate ZIP Codes and Tax Rate Boundaries

task: Investigate whether ZIP codes are an appropriate tool for determining sales tax rates. For example, how frequently do ZIP codes change? Does our proposed use of the first three, five, and all nine digits of ZIP codes correspond well to tax rate jurisdictions? In particular, how large will the ZIP code table be if current jurisdictions do not change much?

number of teams: 2

team size: 1 person

deadline: 02 Mar

sources: library, government publications, WWW

related tasks: 2.3

2.3 Collect Tax Rates

task: Collect sales tax rates categorized by ZIP code for a moderately complicated state such as California, Texas, or Washington. Collect state taxes rate for at least two other regions. We will use this data when running our code.

number of teams: 2

team size: 1 person

deadline: 14 Mar

sources: library, government publications, WWW

related tasks: 2.2

2.4 Current Collection Practices

task: Investigate how multi-store businesses compute and pay sales taxes. Does this involve software programs? tax lawyers? a lot of paperwork? Be sure to investigate an interstate organization. This will probably involve a trip to a store headquarters. Obtain information on the costs of collecting sales taxes and complying with sales tax laws. This may involve searching for studies on the costs of sales tax compliance.

number of teams: 2

team size: 2 persons

deadline: 02 Mar

sources: businesses, library, government publications, WWW

related tasks:

2.5 Survey Proposals to Simplify Sales Taxes

task: Read a majority of available sales tax simplification proposals, summarizing the main ideas and looking for ideas that we omitted from our proposal. In other words, critique proposals.

number of teams: 2

team size: 1 person

deadline: 02 Mar

sources: library, government publications, WWW

related tasks:

2.6 Numbering Items

task: Investigate schemes to number items for sale. What schemes currently exist? What schemes can be extended? Can UPC or EAN codes be used? What problems exist with using UPC codes? How does one obtain lists of UPC codes? Are there already existing channels moving UPC codes from manufacturers to retailers? Will US or UN industry codes really suffice? How many different items are for sale in the United States every year and what fraction of the items changes every year?

number of teams: 2

team size: 1 persons

deadline: 02 Mar

sources: talk to Jeffrey, library, government publications, WWW

related tasks: 1.3