Assignment Week 11: Book Search

Application

In this assignment you will read in a text file; use exception handlining; and use lists of tuples.

Background

We often have to parse in and search data sets. This allows us to make decisions and see patterns that are hard to see by just looking at the raw data.

Often when getting data, it will be in several different formats. CSV (Comma Separated Values) files are the most common. However, even though it is called comma-separated, it often uses other delimiters, such as colons (:), semi-colon (;), or even spaces. Another type of delimiter that can be used are tabs, however this can also be considered a TSV (Tab Separated Values) file. Regardless of the extension (letters after the period), the delimiter is the most important.

One other common formatting feature is that the first line of the file normally contains header information and is not parsed (converted from text to programming data structures). Data structures are ways of organizing data in a way that computers can use the data more easily. A list is a data structure. You will need to handle this parsing of data to a data structure in your solution.

In this application, you will read in a tsv file of books on programming and computing with their ISBN number, list of authors and price. This is a great list of books which you should probably read.

You will then allow the user to search across several of the fields and display the matching results. Users will need to be able to search in the title, ISBN, author, or price. For the results table, you will need to report how many results were found and the average price for all in the table.

Example will be provided below.

Instructions

For this assignment you will create a module and a script which will use the functions in your module.

You will create a list of lists with each field (data element) parsed to the correct data type from the provided text file (books.tsv).

Data File

1. Download the provided text file (books.tsv) from Moodle and copy it into the same folder as your module and script you will be creating.

Create a Python module:

2. Create a script named parse_file.py

3. Create a function named **get_books**
    a. It accepts a string which contains the relative path to the file.
    b. The function will then parse the text file by reading in every line and storing all values in a list of tuples.
        i. You will need to skip the first line as it is just headers.
        ii. Do not forget to convert appropriate field values to the appropriate type.
        iii. You need to return the fields as they are in the provided text file.
            1. Title, ISBN, Author(s), Price
            2. E.g. ['Coders at Work','978-1430219484','Peter Seibel',28.1]
        iv. Do not add additional elements to the tuple.
            1. Each tuple should only have 4 elements.
            2. You are only returning the parsed data from the file.
    c. Make sure to use try:except blocks as appropriate.
        i. On failure, make sure to return an empty list.
    d. Return the list of tuples.

4. Create a function named **get_average_price**
    a. It accepts a list of books as a list..
    b. Using the appropriate value for each item in the list, you will need to calculate average price of all books in the list.
    c. Return the float value of the average.
        i. Do not round the value.
    d. Do not handle case if division by zero happens, allow the exception to be raised.

5. Create a function named **filter_by_title**
    a. It accepts a list of books as lists and search string.
    b. Using the appropriate value for each item in the list, you will need see if the search string is found in the book title.
        i. Hint 1: This might be a good use of the in operator.
        ii. Hint 2: Remember it is a good idea convert your text so you can search it case-insensitive (so that case does not matter).
    c. Return a new list with a copy of all results that match the search criteria.
        i. Do not modify the original list.

6. Create a function named **filter_by_isbn**
    a. It accepts a list of books as lists and search string.

   b. Using the appropriate value for each item in the list, you will need see if the search string is found in the ISBN.
      i. Hint 1: This might be a good use of the in operator.
   c. Return a new list with a copy of all results that match the search criteria.
      i. Do not modify the original list.

7. Create a function named **filter_by_author**
   a. It accepts a list of books as lists and search string.
   b. Using the appropriate value for each item in the list, you will need see if the search string is found in the author(s) names.
      i. Hint 1: This might be a good use of the in operator.
      ii. Hint 2: Remember it is a good idea convert your text so you can search it case-insensitive (so that case does not matter).
   c. Return a new list with a copy of all results that match the search criteria.
      i. Do not modify the original list.

8. Create a function named **filter_by_price**
   a. It accepts a list of books as lists, price to compare against, and a Boolean as to whether to filter by minimum or maximum price.
   b. Using the appropriate value for each item in the list, you will need see if the price is within the range.
      i. If you are searching by maximum price, return all books which are less than or equal to the price given.
      ii. If you are searching by minimum price, return all books which are greater than or equal to the price given.
      iii. Hint 1: Your data should already be in the correct data types in your list.
   c. Return a new list with a copy of all results that match the search criteria.
      i. Do not modify the original list.

Create a Python script:

1. Create a script named book_search.py
2. Get the full results from the file.
3. Store the tax_rate for the calculating price with tax.
   a. We are in Idaho, so use 6% sales tax.
4. Prompt the user for which to search.
   a. Hint: a sentinel loop would help here.
   b. The options must be the following letters:
      i. t – title
      ii. i – ISBN
      iii. a – author names
      iv. x – maximum price
      v. n – minimum price
      vi. l – list all

5. Validate that the user input a correct value.
    a. If the value is not a valid option, prompt the user again.
    b. Only continue when you have a valid option.

6. If the user selected anything other than all, prompt the user for the text to search for and then call the appropriate function to get the filtered list.
7. If the user selects ‘all’, then the results will include all of the books.

8. Display the number of results in the list (filtered or unfiltered as appropriate).

9. Using the list, display a table with has the column headers and the following columns:
    a. Row number
        i. Number of the row, not the original number in the text file.
    b. Book Title
        i. Title of the book.
    c. ISBN
        i. ISBN in the format XXX-XXXXXXXXXX
    d. Author Names
        i. List of authors’ names separated with commas with more than one author.
    e. Price (without tax)
        i. As US dollars
    f. Price (with tax)
        i. As US dollars
10. Finally, using the get_average_price function, get and print out the average price for the list printed (not the whole list).
    a. Use a try:except block so that if the list has no items and has an error, the score is printed as “--.--”
        i. Hint: nothing in the list will cause a divide by zero exception.

Hints:

Start by getting and printing out all results, before you worry about filtering results.

Make sure to test all of your functions, one at a time.

Start small and build on a little bit as you go.

As with every other assignment, the functions listed are the minimum requirements. It might be wise to create extra functions to make your coding easier.

Output

The output should easily readable. As with every other assignment, your output does not need to look exactly like mine. I encourage you to be creative and descriptive in your output. Here is some example output:

```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > t
What should I look for in the title? l
I found 15 results:
#   Book Title                                                        ISBN        Author Name(s)                 Price   w/tax
================================================================================================================================
1   HTML and CSS: Design and Build Websites                      978-1118008188   Jon Duckett                   $17.74  $18.80
2   Beginning Programming All-In-One Desk Reference For Dummies  978-0470108543   Wallace Wang                  $24.12  $25.57
3   Elements of Programming Interviews                           978-1479274833   Adnan Aziz, Tsung-Hsien Lee   $19.95  $21.15
4   Code: The Hidden Language of Computer Hardware and Software  978-0735611313   Charles Petzold               $14.85  $15.74
                                   - - -  (Truncated for brevity, you will print all) - - -
12  Information Technology Essentials Volume 1                   978-1708175146   Eric Frick                    $15.12  $16.03
13  Computer Science Distilled                                   978-0997316025   Wladston Ferreira Filho       $17.39  $18.43
14  Programming Pearls                                           978-0201657883   Jon Bentley                   $40.49  $42.92
15  Working Effectively with Legacy Code                         978-0131177055   Michael Feathers              $53.21  $56.40
```

```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > h
I did not understand; what do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > ti
I did not understand; what do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > t
What should I look for in the title? zzz
I found 0 results:
#   Book Title      ISBN        Author Name(s)   Price   w/tax
===============================================================
---------------------------------------------------------------
                                 Average price: $--.--
```

```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > t
What should I look for in the title? p
I found 14 results:
#   Book Title                                                        ISBN        Author Name(s)                 Price   w/tax
===============================================================================================================================
1   Beginning Programming All-In-One Desk Reference For Dummies  978-0470108543   Wallace Wang                  $24.12  $25.57
2   Beginning C++ Through Game Programming                       978-1435457423   Michael Dawson                $23.47  $24.88
3   Elements of Programming Interviews                           978-1479274833   Adnan Aziz, Tsung-Hsien Lee   $19.95  $21.15
4   Code: The Hidden Language of Computer Hardware and Software  978-0735611313   Charles Petzold               $14.85  $15.74
5   The Principles of Object-Oriented JavaScript                 978-1593275402   Nicholas C. Zakas             $13.29  $14.09
6   How Linux Works: What Every Superuser Should Know            978-1593275679   Brian Ward                    $20.86  $22.11
7   The Psychology of Computer Programming                       978-0932633422   Gerald M. Weinberg            $44.95  $47.65
8   The Pragmatic Programmer                                     978-0135957059   David Thomas, Andrew Hunt     $42.31  $44.85
9   Peopleware: Productive Projects and Teams                    978-0321934116   Tom DeMarco, Tim Lister       $40.49  $42.92
10  Raspberry Pi User Guide                                      978-1119264361   Eben Upton                    $15.00  $15.90
11  The Self-Taught Programmer                                   978-0999685907   Cory Althoff                  $20.78  $22.03
12  Code Complete (Developer Best Practices)                     978-0735619678   McConnell Steve               $23.23  $24.62
13  Computer Science Distilled                                   978-0997316025   Wladston Ferreira Filho       $17.39  $18.43
14  Programming Pearls                                           978-0201657883   Jon Bentley                   $40.49  $42.92
-------------------------------------------------------------------------------------------------------------------------------
                                                                                                 Average price: $25.80
```

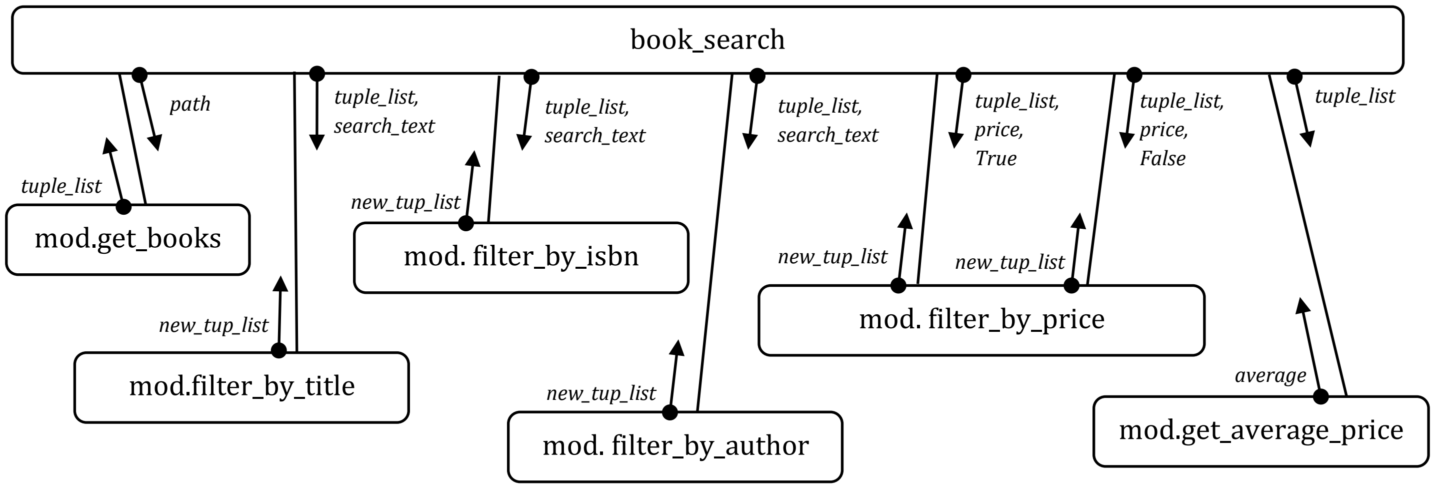
```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > i
What is part of the ISBN? 1435
I found 1 results:
#   Book Title                                   ISBN        Author Name(s)    Price   w/tax
=============================================================================================
1   Beginning C++ Through Game Programming  978-1435457423   Michael Dawson   $23.47  $24.88
---------------------------------------------------------------------------------------------
                                                               Average price: $23.47
```

```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > a
What is part of the author's name? wein
I found 1 results:
#   Book Title                                   ISBN        Author Name(s)        Price   w/tax
=================================================================================================
1   The Psychology of Computer Programming  978-0932633422   Gerald M. Weinberg   $44.95  $47.65
-------------------------------------------------------------------------------------------------
                                                                   Average price: $44.95
```

```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > x
What is the maximum price? 17.00
I found 4 results:
#   Book Title                                                        ISBN        Author Name(s)       Price   w/tax
=====================================================================================================================
1   Code: The Hidden Language of Computer Hardware and Software  978-0735611313   Charles Petzold     $14.85  $15.74
2   The Principles of Object-Oriented JavaScript                 978-1593275402   Nicholas C. Zakas   $13.29  $14.09
3   Raspberry Pi User Guide                                      978-1119264361   Eben Upton          $15.00  $15.90
4   Information Technology Essentials Volume 1                   978-1708175146   Eric Frick          $15.12  $16.03
---------------------------------------------------------------------------------------------------------------------
                                                                                       Average price: $14.56
```

```
What do you want search by:
        (t)itle
        (i)sbn
        (a)uthor
        ma(x) price
        mi(n) price
        or get a(l)l
        > n
What is the minimum price? 40
I found 5 results:
#   Book Title                                      ISBN        Author Name(s)               Price   w/tax
===========================================================================================================
1   The Psychology of Computer Programming     978-0932633422   Gerald M. Weinberg          $44.95  $47.65
2   The Pragmatic Programmer                   978-0135957059   David Thomas, Andrew Hunt   $42.31  $44.85
3   Peopleware: Productive Projects and Teams  978-0321934116   Tom DeMarco, Tim Lister     $40.49  $42.92
4   Programming Pearls                         978-0201657883   Jon Bentley                 $40.49  $42.92
5   Working Effectively with Legacy Code       978-0131177055   Michael Feathers            $53.21  $56.40
-----------------------------------------------------------------------------------------------------------
                                                                             Average price: $44.29
```

Structure chart



Scoring

1. 5% - Compiles without errors

parse_file.py

2. 18% - get_books (type hinting, correct parameters)
    a. uses try:except, opens file, reads lines correctly, parses data into tuple, uses correct data types, returns list of tuples, returns empty list on exception
3. 5% - get_average_price (type hinting, correct parameters)
    a. returns correct average from lists
4. 10% - filter_by_title (type hinting, correct parameters)
    a. iterates each inner list, checks for search text in value, uses correct values, returns new list
5. 9% - filter_by_isbn (type hinting, correct parameters)
    a. iterates each inner list, checks for search text in value, uses correct values, returns new list
6. 10% - filter_by_author (type hinting, correct parameters)
    a. iterates each inner list, checks for search text in value, uses correct values, returns new list
7. 17% - filter_by_price (type hinting, correct parameters)
    a. Max - iterates each inner list, checks for price in value, uses correct values, returns new list of books with a cost less than the price if max price is selected
    b. Min - iterates each inner list, checks for price in value, uses correct values, returns new list of books with a cost greater than the price if min price is selected

grader. py

8. 1% - calls get_books
9. 1% - prompts user for input
10. 2% - loops until correct answer
11. 3% - prompts for search input correctly
12. 3% - calls correct function based on type of search
13. 1% - displays number of results
14. 1% - displays table header from filtered list
15. 5% - displays results in a table
16. 2% - displays average of results in table
17. 2% - uses try:except to handle for average of empty list
18. 10% - Good Coding (compiles, good comments, header block, doc_strings, comments andgood variable names)