

Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

11.7 "Restore" menu option

For this option, you'll be implementing a new function, `load_tasks_from_csv()`. This function

- reads data from a csv file, one line at a time
- validates it by trying to create a task
- stores that data into the list of tasks.

“ Advice: If you have trouble with your `load_tasks_from_csv()`, we recommend reviewing Section 10.6, and 10.7 on zyBooks.

To implement your function, you must `import csv` in the `task_functions.py` file. For more information, refer to zyBooks section 10.7. You can also refer to [LA 10.8](#).

The template for `load_tasks_from_csv()` is provided below.

```
def load_tasks_from_csv(filename, in_list, priority_map):  
    """  
    param: filename (str) - A string variable which represents  
    the  
        name of the file from which to read the contents.  
    param: in_list (list) - A list of task dictionary objects  
    to which  
        the tasks read from the provided filename are  
    appended.  
        If in_list is not empty, the existing tasks are not  
    dropped.  
    param: priority_map (dict) - a dictionary that contains the  
    mapping  
        between the integer priority value (key) to its  
    representation  
        (e.g., key 1 might map to the priority value  
    "Highest" or "Low")  
        Needed by the helper function.  
  
    The function ensures that the last 4 characters of the  
    filename are '.csv'.
```

The function requires the ``import csv`` and ``import os``.

If the file exists, the function will use the ``with`` statement to open the ``filename`` in read mode. For each row in the csv file, the function will

proceed to create a new task using the ``get_new_task()`` function.

- If the function ``get_new_task()`` returns a valid task object,

it gets appended to the end of the ``in_list``.

- If the ``get_new_task()`` function returns an error, the 1-based

row index gets recorded and added to the NEW list that is returned.

E.g., if the file has a single row, and that row has invalid task data,

the function would return `[1]` to indicate that the first row caused an

error; in this case, the ``in_list`` would not be modified.

If there is more than one invalid row, they get excluded from the

`in_list` and their indices will be appended to the new list that's

returned.

returns:

- * `-1`, if the last 4 characters in ``filename`` are not ``.csv``
- * `None`, if ``filename`` does not exist.
- * A new empty list, if the entire file is successfully read from ``in_list``.

- * A list that records the 1-based index of invalid rows detected when

calling `get_new_task()`.

Helper functions:

- `get_new_task()`

"""

Sample task files

Below are the contents of the files that are used in the sample program flow below. Make sure that there are no empty lines, otherwise, that row is counted as being invalid.

missing-date.csv

```
Finish CSW8 LAs,Finish LA 10 to prepare for the final
project,5,,no
```

combined.csv

```
Finish CSW8 LAs,Finish LA 10 to prepare for the final
project,5,06/02/2022,no
,Finish LA 10 to prepare for the final project,5,06/02/2022,no
duplicate task,,5,06/02/2022,no
Finish CSW8 LAs,Finish LA 10 to prepare for the final
project,,06/02/2022,no
Finish CSW8 LAs,Finish LA 10 to prepare for the final
project,5,m/d/y,no
Finish CSW8 LAs,Finish LA 10 to prepare for the final
project,5,06/02/2022,nope
```

single-task.csv

```
Finish CSW8 LAs,Finish LA 10 to prepare for the final
project,5,06/02/2022,no
```

Sample Program Walkthrough

The corresponding code for the **main program** is similar to the one for the S. Your task to figure out what to add using the following user interactions. As always, the ellipses are **NOT** part of the code.

```
...
You selected option R to > Restore data from file.
::: Enter the filename ending with '.csv'.
> test.csv
WARNING: |test.csv| was not found!
::: Would you like to try again? Enter 'y' to try again.
> y
::: Enter the filename ending with '.csv'.
> missing-date.csv
WARNING: |missing-date.csv| contains invalid data!
The following rows from the file were not loaded:
[1]
::: Would you like to load another file? Enter 'y' to try
again.
> y
::: Enter the filename ending with '.csv'.
> combined.csv
WARNING: |combined.csv| contains invalid data!
The following rows from the file were not loaded:
[2, 4, 5, 6]
::: Would you like to load another file? Enter 'y' to try
again.
```

```
> y
::: Enter the filename ending with '.csv'.
> single-task.csv
Successfully stored all tasks to |single-task.csv|
::: Press Enter to continue
...
```

NOTE: If your Save the data to file option is working correctly, the file you wrote into using that option should work perfectly with this option and your entire task information should be loaded properly.

398092.2571084.qx3zqy7

**LAB
ACTIVITY**

11.7.1: "Restore" menu option

0 / 1



main.py

1

Develop mode

Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

Run program

Input (from above)



main.py
(Your program)

Program output displayed here

Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

[Trouble with lab?](#)