

CS 2316

Individual Homework 5 - Hours in a Day

Due: Monday September 29th, 2014

Out of 100 points

Files to submit: 1. HW5.py

This is an INDIVIDUAL assignment!

Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 2316, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:

- TA Helpdesk - Schedule posted on class website.
- Email TA's or use T-Square Forums

-Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).

-Do not wait until the last minute to do this assignment in case you run into problems.

Premise:

In this assignment, you will be tasked with reading data from files in CSV format. You will be provided with CSV files that contain information about the number of hours a person has planned to spend on four activities (studying, exercising, eating, and socializing) every day for a certain number of days. From this information, you will need to determine, on average, the busiest day of the week, the most time consuming activity, and the number of days the person has planned to go without sleep.

In order to do so, you will be required to write the following methods in a class:

1. `__init__`
2. `importCSV`
3. `busiestDay`
4. `consumingActivity`
5. `sleeplessDays`
6. `calculate`

File Format Information

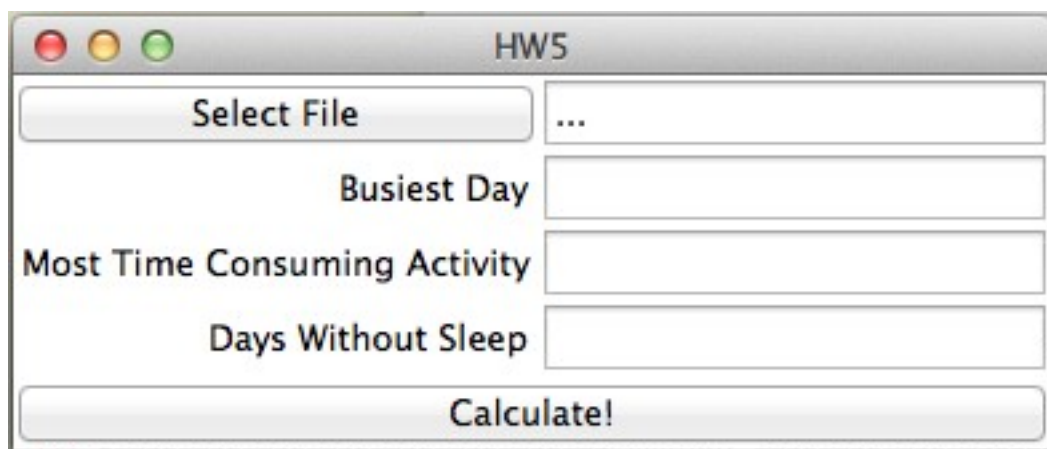
For this assignment, you will be given two comma separated value (CSV) files. These files will have different numbers of rows/columns,

but the format will generally be the same. Each row starts with a day of the week and each column starts with the name of an activity (studying, exercising, eating, socializing). The other values in the table are numbers representing the number of hours planned to be spent in the given day on the corresponding activity. Some days may have more than 24 hours worth of activities planned. See the week.csv, semester.csv and year.csv files for examples. The week.csv file contains the activity plan for 1 week (7 days). The semester.csv file contains the activity plan for one GT semester, 15 weeks (105 days). The year.csv file contains the activity plan for one full GT school year, 41 weeks (287 days).

Your program must work with and produce the correct answers for any similarly formatted data file, regardless of size. You are allowed to use the csv module to read this data, or you may write your own function to read the data.

Function Name: **`__init__`**

This method is automatically called whenever you create a new instance of your object. The `__init__` method is responsible for arranging and initializing your GUI. You should create a GUI that looks like the following:



Note that the labels and text entry widgets are carefully arranged:

- The window is titled
- The labels are right ("East") justified
- The "Select File" button spans the entire left side of the frame
- The filename display text entry widget starts out with "..." in it
- All of the text entry widgets are set to "readonly"

We suggest you use a grid layout to achieve this effect, although if you are able to do so by using multiple frames and the "pack" layout manager you may use that instead.

The "Select File" button should call the importCSV method. The "Calculate!" button should call the calculate method

Function Name: **importCSV**

Parameters:

None

Return Value:

None

Instance Variable:

self.dataList- List containing each row of CSV file.

Description:

This method should read in the file "week.csv", "semester.csv", or "year.csv" by using the CSV module. It should then store the data in the instance variable "self.dayList". Note that all information in the CSV is of type String to begin with so you will need to convert all numerical data (hours) to int or float types.

Function Name: **busiestDay**

Parameters:

None

Return Value:

None

Instance Variables:

self. dataList

Description:

This method will use the self.dataList to determine the busiest day of the week and update the appropriate entry in the GUI. The busiest day of the week will be the day with the highest average number of hours planned. For example, the average number of hours planned for Monday would be the total number of hours planned for each Monday in the CSV file, divided by the total number of Mondays in the CSV file.

Function Name: **consumingActivity**

Parameters:

None

Return Value:

None

Instance Variables:

self. dataList

Description:

This method will use the self.dataList to determine the most time consuming activity and update the appropriate entry in the GUI.

The most time consuming activity will be the activity with the highest average number of hours planned (total hours planned for a given activity divided by the total number of days).

Function Name: **sleeplessDays**

Parameters:

None

Return Value:

None

Instance Variables:

self. dataList

Description:

This method will use the self.dataList to determine the number of days the person has planned to go without sleep (any days where the sum of the hours the person has planned for activities is greater than or equal to 24) and update the appropriate entry in the GUI. For example, if a row of the CSV file looks like:

'Monday', '10', '9', '4', '4'

Then that particular day counts as a day without sleep because there were 27 hours of activities planned.

Function Name: **calculate**

Parameters:

None

Return Value:

None

Description:

This method will call the busiestDay, consumingActivity, and sleeplessDays methods and should result in the GUI being populated. Note the TA should be able to import a new CSV file and click the "Calculate!" button again without the program losing functionality.

Grading:

You will earn points as follows for each piece of functionality that works correctly according to the specifications

The GUI	20
The GUI has all required components	5
All buttons and entries work properly	15
importCSV()	10
Properly reads in csv file	5
Makes data accessible to other methods	5
busiestDay()	20
• Tracks the average hours planned/day	5
• Determines the most time consuming activity	10
• Updates the GUI appropriately	5
consumingActivity()	20
• Tracks the average hours spent on each activity	5
• Determines the most time consuming activity	10
• Updates the GUI appropriately	5
sleeplessDays()	20
• Properly determines the number of sleepless days	15
• Updates the GUI appropriately	5
calculate()	10
• Calls all required functions	3
• GUI updated when pressed	4
• May be pressed multiple times	3