

## Timed Lab 2 – Access Analysis

This is a Timed Lab; this Timed Lab is worth 26 **Exam** points.

For this Timed Lab, you <i>may</i> use	However, you <i>may not</i>
<ul style="list-style-type: none"> <li>• Course notes</li> <li>• Homeworks</li> <li>• Recitation assignments</li> <li>• Other course material</li> <li>• Any material you may find on the Internet that don't involve communicating "live" with other people.</li> </ul>	<ul style="list-style-type: none"> <li>• Communicate with other people/students in real-time via any means. This means no Facebook, email, Piazza, IM, IRC, cell phones, Google Talk, smoke signals, etc.</li> <li>• Share code with other students.</li> <li>• Look at other students work.</li> </ul>

The TAs will be available to answer clarifying questions about the problem, but they are not permitted to give assistance with debugging code, program logic, etc. You will have an entire recitation period to work on this assignment; this time begins *exactly* when your recitation begins and ends *exactly* when your recitation ends: No extra time will be given if you arrive late, except in highly extenuating circumstances that must be approved by Dr. Summet.

T-Square will not permit any late submissions; ensure that you submit your code to T-Square several times to prevent earning a zero due to you being unable to submit. Your TAs will give a verbal warning 10 and 5 minutes before the end of the recitation period; you should submit at these times. If you are taking this timed lab out of section (with approval from Dr. Summet or the Head TA), please e-mail your code to your grading TA by the end of the recitation you are in. **Modifying your code after you leave the recitation room will result in a grade of zero.**

In your collaboration statement, if you use code from somewhere that is *not* a class resource (i.e. not listed on the course calendar), please list where this code came from. Ensure that you fill out the header at the top of the file.

### Problem Description:

You have been asked to consult on a job analyzing foot traffic through a particular building's various secured doors in order to determine whether additional entry or exit points need to be added. However, the data set you have been given is far too massive to sort through by hand, as this is an extremely high traffic building! You've decided that you'll use Python to help you analyze this data. Specifically, you'll be creating a very simple GUI that will help you determine the total amount of traffic over a particular hour for either all the entrances or exits in the building or just one particular entrance or exit.

You may organize your code however you wish, but splitting the problem into a class and a few methods might prove to be helpful.

### Data Format:

The data about the uses of each entrance/exit is given to you in the following format:

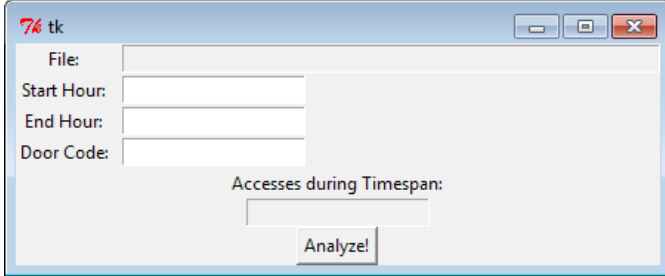
MM/DD/YY|HH:MM|DOOR CODE|ACCESS CARD NUMBER

Each piece of data in the file is separated by a vertical pipe | (Shift + / on a standard US QWERTY keyboard layout). The first piece of data is the date of the access in MM/DD/YY format (ex. 02/21/12). The second piece of data is the Hour and Minute of the access, in 24-hour format (ex. 15:05). The third piece of data is the door code which describes which door was accessed. The door code is simply a string, such as CCB-1S or ABC1234. The last piece of data is the access card number of the person who opened the door. This is also a string, such as 1234567.

For the purposes of this program, only the time and door code are important to you.

## The GUI:

This is a potential gui, but your GUI can be arranged however you want, as long as all of the specified elements (labels, entries, and button) exist and the state (readonly or normal) is correct.



Note that the file name entry box is read only and should be width=60. The Accesses during Timespan entry box is also read only.

When the user clicks the Analyze! button, you should display an askopenfilename dialog box to prompt the user to select an input file. You may assume the user will select a valid file. When the user does select a file, you should put that filename into the File entry box. You should then run your analysis based on what the user typed into the other three entry boxes.

## The Analysis:

For the analysis of each file, you should only count entries that have an hour which is greater than or equal to the start hour and less than the end hour. For example, if the user entered 13 into the Start Hour box and 14 into the End Hour box, your program would count all accesses between 13:00 and 13:59, but would not include 14:00. You may assume that the Start Hour and End Hour entry boxes will always have a valid hour (as an integer) before the user presses the Analyze button.

If there is text in the Door Code box, you should only count accesses through a door which has a Door Code that matches the door code entered in the entry box during the given timespan. If there is no text in the Door Code entry box, you should count accesses through all doors during the given timespan.

Once you have finished counting all appropriate accesses, you should put the total number of accesses you've counted into the Accesses during Timespan box.

## Grading:

- +3: All labels present in GUI (-1 for each missing label)
- +3: Start Hour, End Hour, and Door Code entry boxes present (-1 for each missing)
- +3: File and Accesses during Timespan entry boxes present, readonly, (-1.5 for each missing, -1 for each not readonly).
- +1: Analyze button exists
- +1: Analyze button triggers analysis code when clicked
- +2: Correctly uses askopenfilename dialog (+1 for valid attempt, +1 for correct usage)
- +1: Correctly places filename into File entry box
- +1: Correctly opens file
- +1: Correctly closes file
- +3: Correctly gets data out of file (+1 for valid attempt)
- +3: Correctly considers Start & End Hour (+1 for valid attempt, +1 for correctly considering start hour, +1 for correctly considering end hour)
- +3: Correctly considers Door Code (+1 for valid attempt, +1 for no door code specified, +1 for door code specified)
- +1: Correctly places total accesses into entry box