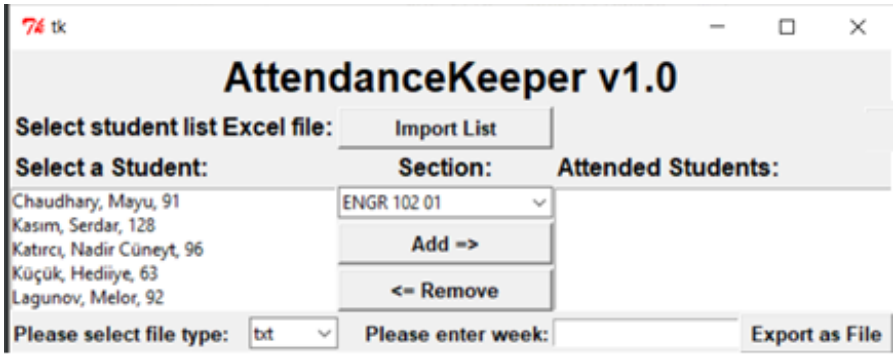
ENGR 102 – Programming Practice
Mini Project 1
Spring 2018

Tags: Tkinter, Events, Data Parsing, Class, Object, Method, GUI Widgets, Exceptions

Keeping track of attendance in a manual way, especially in large classes, may be a big hassle for the teaching staff. In this mini project, you are going to develop a software tool to help with attendance record keeping for a class. Please read on more details.

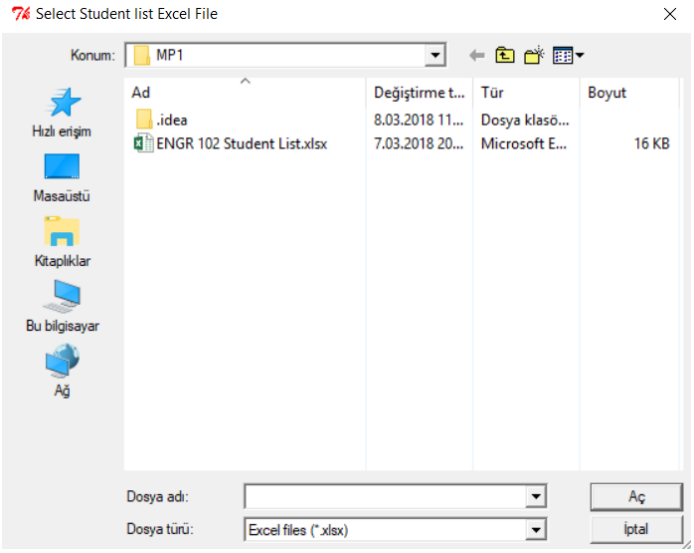
How should it look like?

The graphical user interface of your tool should look like as the following one.



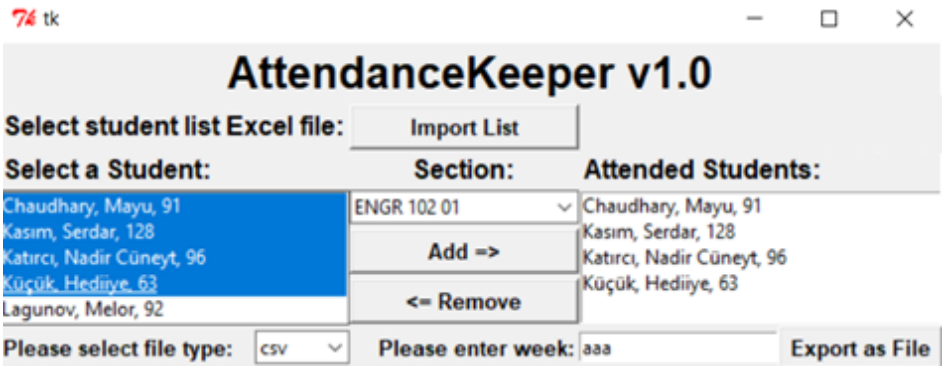
How should it work?

1. First, the user should load the list of students in the class. When the user presses the "Import List" button, the program should open a file navigation dialogue, and ask for a student list in the form of a custom Excel file. A sample excel file is provided as part of the project documents to test your code.



2. At the end of the first step, the student list data should now be now loaded, Next, the user will select a section from a Combobox widget, and the listbox on the left should be populated with the students of the selected section automatically. That is, if user selects "ENGR 102 10", your program should only display students from 10th section of ENGR 102. If the user selects another section, the new section's students should be loaded into the listbox. The loaded data should be in form of *"Surname, Name, ID"* and alphabetically ordered. The default selection in the section combobox should be "ENGR 102 01".

3. User can select one or **multiple** students from the student list, and click on the "Add" button to add the selected students into "Attended students" listbox on the right. If the user changes section at any time through the combobox, the contents of the listbox on the right should also be cleared.



4. Finally, the user should be able save the attendance records. First, the user chooses a file type via a combobox. The available file types should be "xls", "csv", and "txt" (by default, "txt" should be selected). When user clicks on "Export as File" button, the program should save the attendance data into a file of selected type. The file should be saved in the current working directory with file name set as *section name* + *Entry widget value* + *file type* (e.g., "ENGR 102 01 Week 2.txt").

   a. If the user selects "csv" file type, you need to **raise a BaseException** with a message, "File type is not supported", as shown below.

```
C:\Python27\python.exe "D:/Personal/Dropbox/ENGR 102 Internal/Mini Projects/MP1/solution.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python27\lib\lib-tk\Tkinter.py", line 1542, in __call__
    return self.func(*args)
  File "D:/Personal/Dropbox/ENGR 102 Internal/Mini Projects/MP1/solution.py", line 232, in export_as_file
    raise err
BaseException: File type is not supported!
```

   b. If the user selects ".xls", you should create an excel file with columns of "ID", "Name", and "Department" which will hold student ID, student name and surname, and department, respectively.

   c. If the user selects ".txt", you should create text file where on each line should contain the ID, Name-Surname, Department information of a student separated by tab. Remember that name format should be "*Name Surname*".

Implementation Notes:

- **You need to use at least 3 classes in this project. For instance, you may create one class that represents Student, another one for StudentList, and one for GUI. Please design your code accordingly.**
- For an example of Combobox usage, you may take a look at the following link:

  https://stackoverflow.com/questions/17757451/simple-ttk-combobox-demo

- You will need to use event binding to handle some parts of the project. Events will be covered at the beginning of the upcoming project. If you do not want to wait, the above link has an example of it that you may use as a reference.
- You may use **xlrd** module to read Excel files and xlwt to write into Excel files (install them on PyCharm in the same way you did the other modules). The following tutorial have example usages of these modules.

  https://goo.gl/KPm3Sa

- To have the user choose the transcript file from your computer, you may use tkFileDialog. The following link contains many useful examples of using tkFileDialog.

  https://pythonspot.com/tk-file-dialogs/

- To work with ListBox, please see the following reference:

  http://effbot.org/tkinterbook/listbox.htm

  Besides, the listbox should be associated with a scrollbar as well. The following link may be useful.

  http://effbot.org/zone/tkinter-scrollbar-patterns.htm

- Apart from the above, the lecture slides are enough for most parts, still feel free to research and implement. Please provide references if you use outer sources. TutorialsPoint, StackOverflow and Effbot might be good starting points, but **do not copy codes directly**.

Warnings:

- You **CANNOT** use place for geometry, only grid and pack are allowed.
- Do not talk to your classmates on project topics when you are implementing your projects. Do not show or email your code to others. If you need help, talk to your TAs or myself, not to your classmates. If somebody asks you for help, explain them the lecture slides, but do not explain any project related topic or solution. Any similarity in your source codes will have serious consequences for both parties.
- Carefully read the project document, and pay special attention to sentences that involve "should", "should not", "do not", and other underlined/bold font statements.

- If you use code from a resource (web site, book, etc.), make sure that you reference those resource at the top of your source code file in the form of comments. You should give details of which part of your code is from what resource. Failing to do so may result in plagiarism investigation. Last but not the least, you need to understand code pieces that you may get some other resources. This is one of the goals of the mini projects.
- Even if you work as a group of two students, each member of the team should know every line of the code well. Hence, it is important to understand all the details in your submitted code.

How and when do I submit my project?

- Projects may be done individually or as a small group of two students (doing it individually is **strongly** recommended for best learning experience). If you are doing it as a group, only **one** of the members should submit the project. File name will tell us group members (Please see the next item for file naming details).
- Submit your own code in a single Python file. Name it with your and your partner's first and last names. As an example, if your team members are Deniz Barış and Ahmet Çalışkan, then name your code file as deniz_baris_ahmet_caliskan.py (Do not use any Turkish characters in file name). If you are doing the project alone, then name it with your name and last name similar to the above naming scheme.
    - Those who do not follow the above naming conventions will **get** 10% **off** of their project grade.
- Submit it online on LMS by 17:00 on April 9, 2018.

  Late Submission Policy:

  - -10%: Submissions between 17:01 – 18:00 on the due date
  - -20%: Submissions between 18:01 – midnight (00:00) on the due date
  - -30%: Submissions after which are up-to 24 hours late.
  - -50%: Submissions which are up-to 48 hours late.
  - Submission more than 48 hours late will not be accepted.

Grading Criteria?

GUI Design (25)	Loading Class List (20)	Displaying Student List Properly in the Listbox and Sections in the Combobox (20)	Add / Remove Students between Listboxes (10)	Export as File (25)

From your overall grade, we will deduct points by the specified percentage for the following items:

- Inappropriate/Cryptic variable names (10%)
- Classes and objects are not used properly (30%)
- Sufficient commenting (10%).

Have further questions?:

- If you need help with anything, please use the office hours of your TAs and the instructor to get help. **Do not walk in randomly (especially on the last day) into your TAs' or the instructor's offices. Make an appointment first. This is important. Your TAs have other responsibilities. Please respect their personal schedules.**