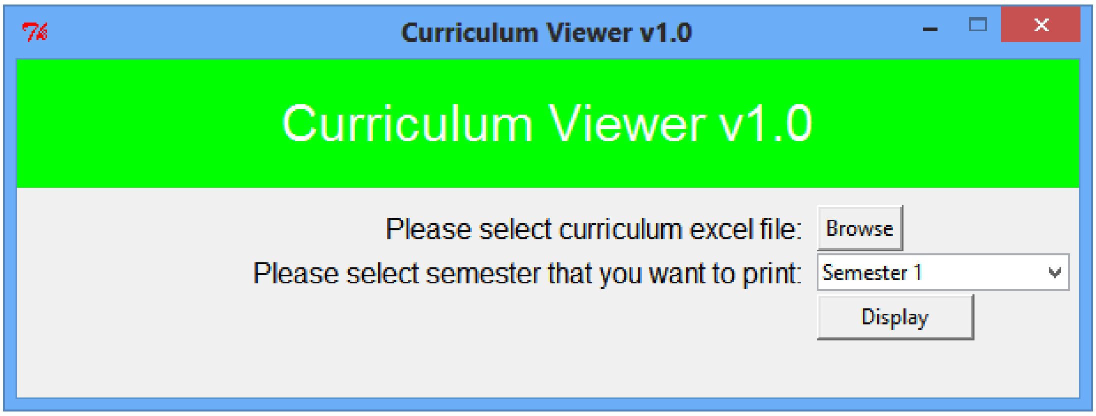
ENGR 212 Programming Practice
Mini Project 1
March 1, 2016

In this mini project, you are going to develop a tool, called *CurriculumViewer*. Details regarding the requirements are as follows:

1. Your program will have a graphical user interface (GUI) which will look like as shown in Figure 1. You may use any coloring-scheme you like. The below one is for illustration purposes only. Details about how it should work are provided below.



Please select curriculum excel file:
Please select semester that you want to print: Semester 1

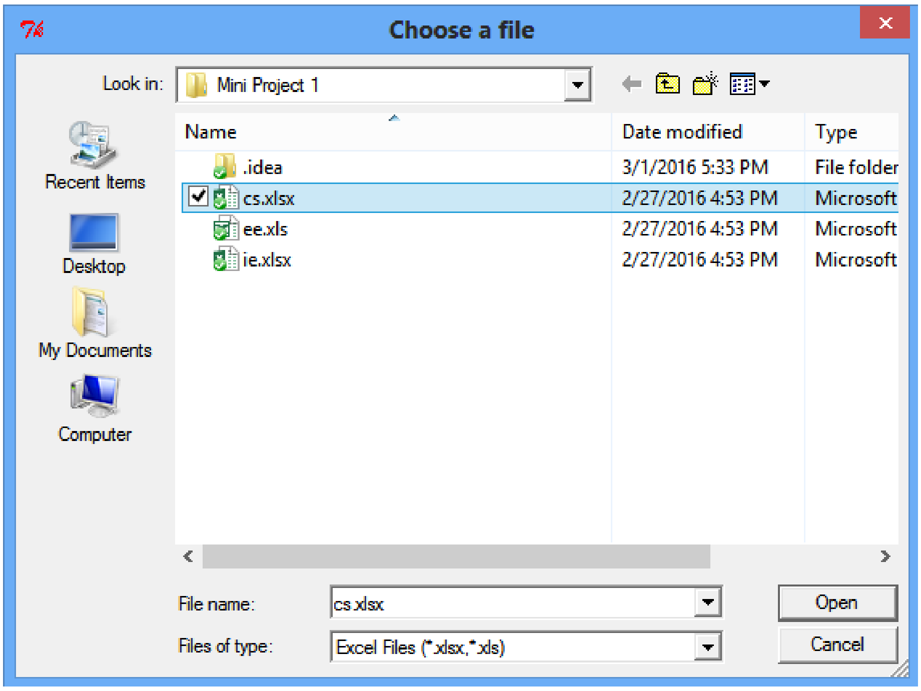
Figure 1

- Your software will accept the curriculum for any SEHIR engineering department in the form of an excel file as input. The user will choose the curriculum file by clicking on 'Browse' button, and a file dialog window will open (see Figure 2). Then, the user will choose an excel file that contains a curriculum of either EE, CS, or IE departments at SEHIR. The Excel file content should be assumed to be formatted as in the example Excel documents uploaded on LMS. We will test your programs with similar or the same files.
- Next, the user will choose a semester to display.
- Finally, the user will click on the button, 'Display', which will display the courses in the selected semester with their credit and code information as shown in Figure 3.

2. After displaying a semester, the user should be able to select another semester, and then click on the "Display" button again to refresh the course list accordingly. Here, to implement the course list, you **should** use a separate label widget for each course name, course code, credit information, and so on, and organize them in a nice way using pack or grid layout managers that we covered. You **should not** use the place geometry manager in any part of the project.
3. Once the user provides a curriculum excel file in step 1, your program should also create a database (anydbm db) which stores curriculum information read from the excel file. When your program is run again for the next time, if the user does not provide a new excel file via clicking the 'Browse' button, your program should still be able to display information from the database that was created in the previous runs. If the user selects a new curriculum file, the database content should be refreshed as well. Database file name should be 'curriculum.db'. If the user tries to click on 'Display' button before selecting a

file, and if there is no previously created database file, your program should warn the user with a message box saying that ‘A curriculum file should be selected first by clicking on the Browse button’. Note that this warning message should not be displayed, if the user loaded a curriculum file in the past runs which should have created a database (curriculum.db) in its current working directory.

4. Write your code in a single Python file.



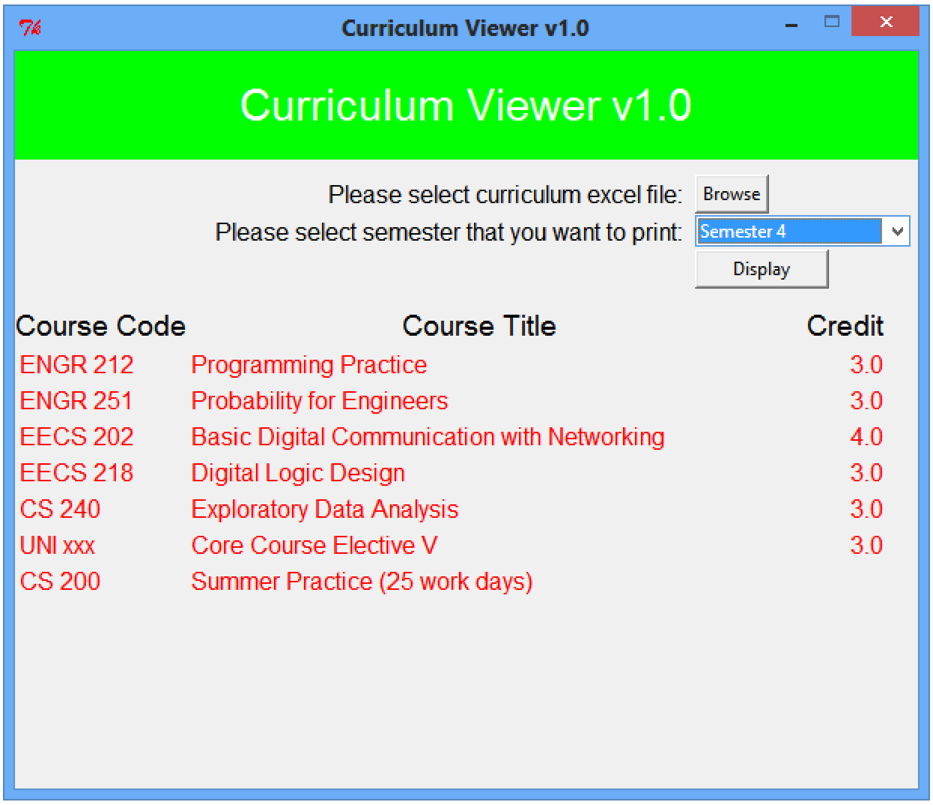
Look in: Mini Project 1

Name	Date modified	Type
.idea	3/1/2016 5:33 PM	File folder
☑ cs.xlsx	2/27/2016 4:53 PM	Microsoft
ee.xls	2/27/2016 4:53 PM	Microsoft
ie.xlsx	2/27/2016 4:53 PM	Microsoft

File name: cs.xlsx

Files of type: Excel Files (*.xlsx,*.xls)

Figure 2



Please select curriculum excel file:

Please select semester that you want to print: Semester 4

Course Code	Course Title	Credit
ENGR 212	Programming Practice	3.0
ENGR 251	Probability for Engineers	3.0
EECS 202	Basic Digital Communication with Networking	4.0
EECS 218	Digital Logic Design	3.0
CS 240	Exploratory Data Analysis	3.0
UNI xxx	Core Course Elective V	3.0
CS 200	Summer Practice (25 work days)	

Figure 3

Can you provide any further pointers that may be helpful? :

- You may use **xlrd** module to read Excel files (install it on PyCharm in the same way you did the other modules). The following tutorial have example usages of this module. The code piece on page 9 of this tutorial should be sufficient for you.

  http://www.simplistix.co.uk/presentations/python-excel.pdf

- Our coverage in the class did not include using a file dialog (tkFileDialog), but it is really simple, and the following link contains many useful examples of using tkFileDialog.

  http://tkinter.unpythonic.net/wiki/tkFileDialog

- See the following link on how to create a simple message dialog box:

  http://www.tutorialspoint.com/python/tk_messagebox.htm

- For semester selection, you may use the ComboBox widget of Tkinter. The following example code piece may help: http://stackoverflow.com/questions/17757451/simple-ttk-combobox-demo

Warnings:

- **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.
- 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. You may be interviewed about any part of your 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 recommended). 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 details).

- Submit your own code in a **single** Python file. Name it with your and your partner's first and last names (see below for naming).
  - 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.

- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on March 15, 2016**.

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 which are 24 hour late.
- -50%: Submissions which are 48 hours late.
- Submission more than 48 hours late will not be accepted.

Grading Criteria? :

- Does it run? (Submissions that do not run will get some partial credit which will not exceed 30% of the overall project grade).
- Does it implement all the features according to the specifications and produce correct results?
- Code organization (Meaningful names, sufficient and appropriate comments, proper organization into functions and classes, clean and understandable, etc.)?
- Interview evaluation (your grade from interview will be between 0 and 1, and it will be used as a coefficient to compute your final grade. For instance, if your initial grade was 80 before the interview, and your interview grade is 0.5, then your final grade will be 80*0.5 = 40). Not showing up for the interview appointment will **result in** grade 0.

Have further questions? :

- Please contact your TAs (Jareth or Dogukan are focusing on projects. You may want to talk to them first, but you may talk to Ali and Bekir as well) if you have further questions. If you need help with anything, please use the office hours of your TAs and the instructor to get help.