ENGR 212 Programming Practice
Mini Project 3
March 29, 2016

During your study at SEHIR, you are offered a number of classes that you need to complete some of them in order to graduate. Have you ever wondered how these courses are related in terms of their coverage? How close or distant are they from each other? In this project, you are going to help new (and maybe existing) students understand how courses are (or not) related to each other by providing them a visualization tool. 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. Details about how it should work are provided below.

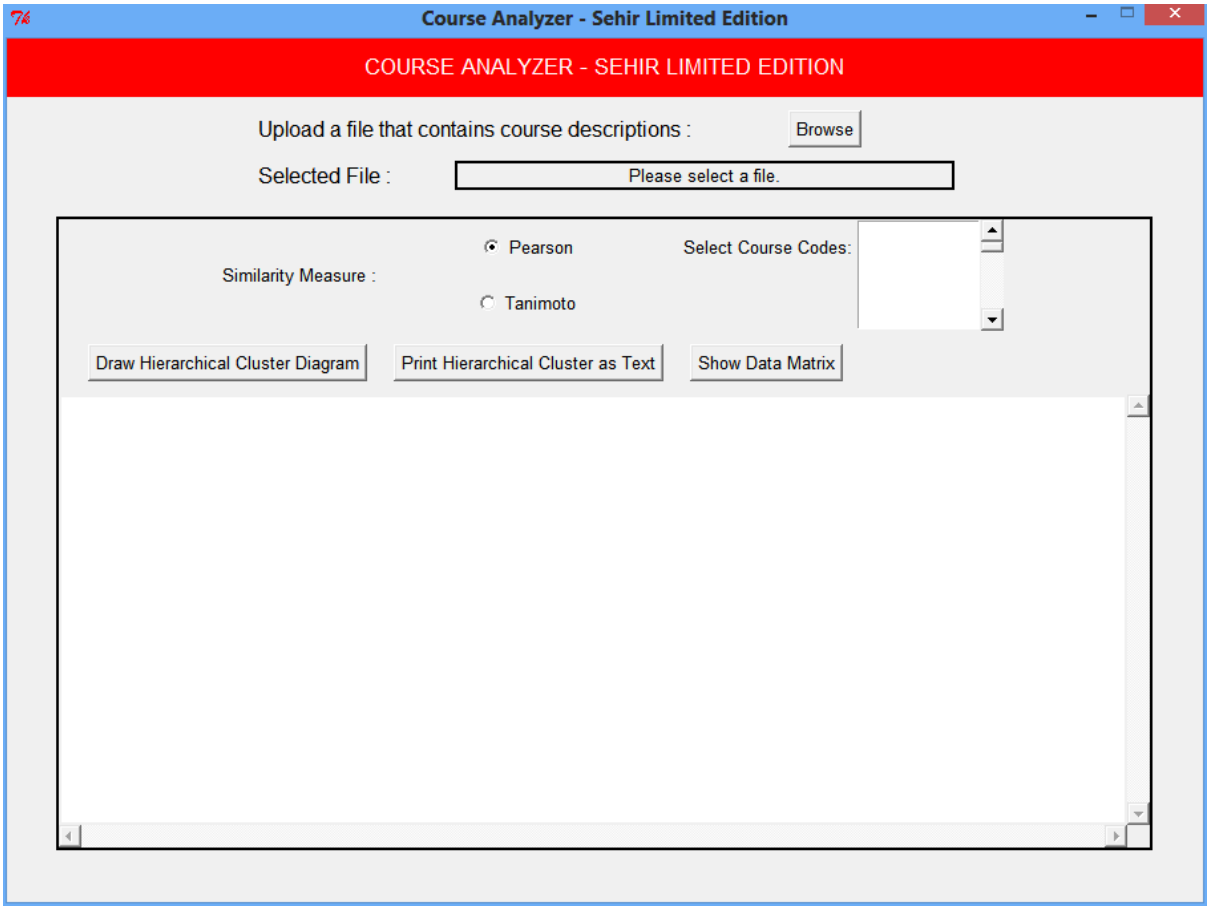

Figure 1

- The user will first upload a text file that contains the titles and descriptions of courses that are offered as part of a particular department's curriculum (sample input files are provided for CS, EE, and IE departments. You may test your software with any of these). Once the user chooses an input file, a label below the browse button shows the full path of the selected file (see Figure 2).
- In the next section, the user will configure settings for the course analyzer. On the left, the user will choose a similarity measure: Pearson or Tanimoto (both of which are provided in clusters.py in the project zip file). On the right hand side, the user will choose the course codes that s/he wants to include in the analysis (e.g., ENGR, EECS, IE, etc.). This widget should be a listbox that allows multiple selection. It will be initially empty (see Figure 1) when your program starts. It should be populated after the user selects a file based on this file's content (basically, you need to extract all unique course codes from the provided file). Course

codes should be sorted alphabetically in the listbox (see Figure 2).

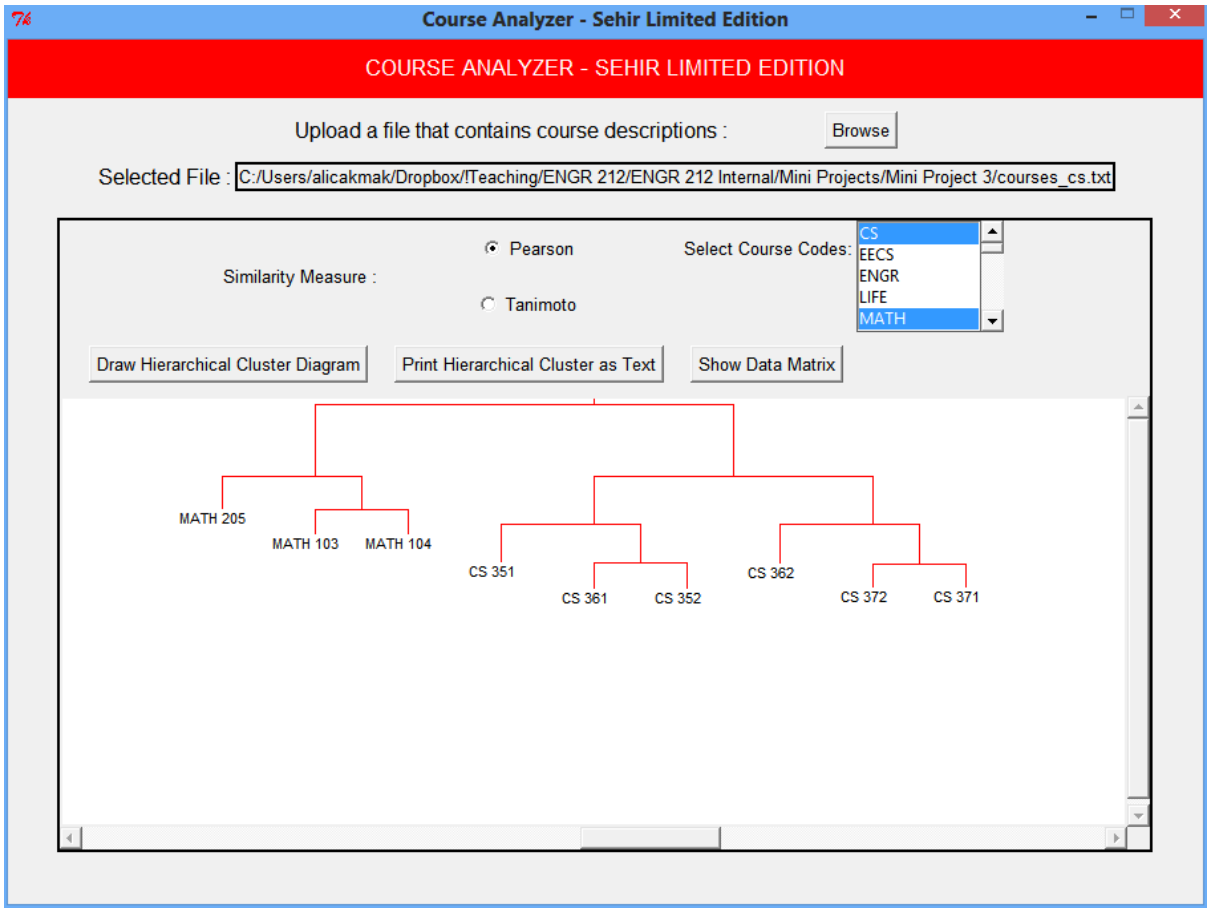

Figure 2

- The bottom part will have a row of three buttons. Below buttons, there will be a canvas with white background.
  - When the first button ("Draw Hierarchical Cluster Diagram") is clicked, your program will draw the result of hierarchical clustering on to canvas. In the class, I showed you how to produce an image file that contains the visualization of the clusters in a hierarchical way using the *drawdendogram* function. Here, you are going to produce a similar visualization on canvas, but you will **not** use *drawdendogram* function. You are going to draw it by yourself using the methods provided by canvas (such as, create_line, create_text, etc.) See Figure 2 for a sample hierarchical cluster drawing on canvas when CS and MATH course codes are chosen by the user. Your canvas should have horizontal and vertical scrollbars. The canvas size should be dynamically set based on the expected size of the hierarchical clustering drawing.
  - When the second button ("Print Hierarchical Cluster as Text") is clicked, your program should print the text representation of the hierarchical clustering result on canvas (see clust2str function in clusters.py) (see Figure 3 as an example)
  - When the third button ("Show Data Matrix") is clicked, your program should print data matrix on canvas (see Figure 4 as an example).

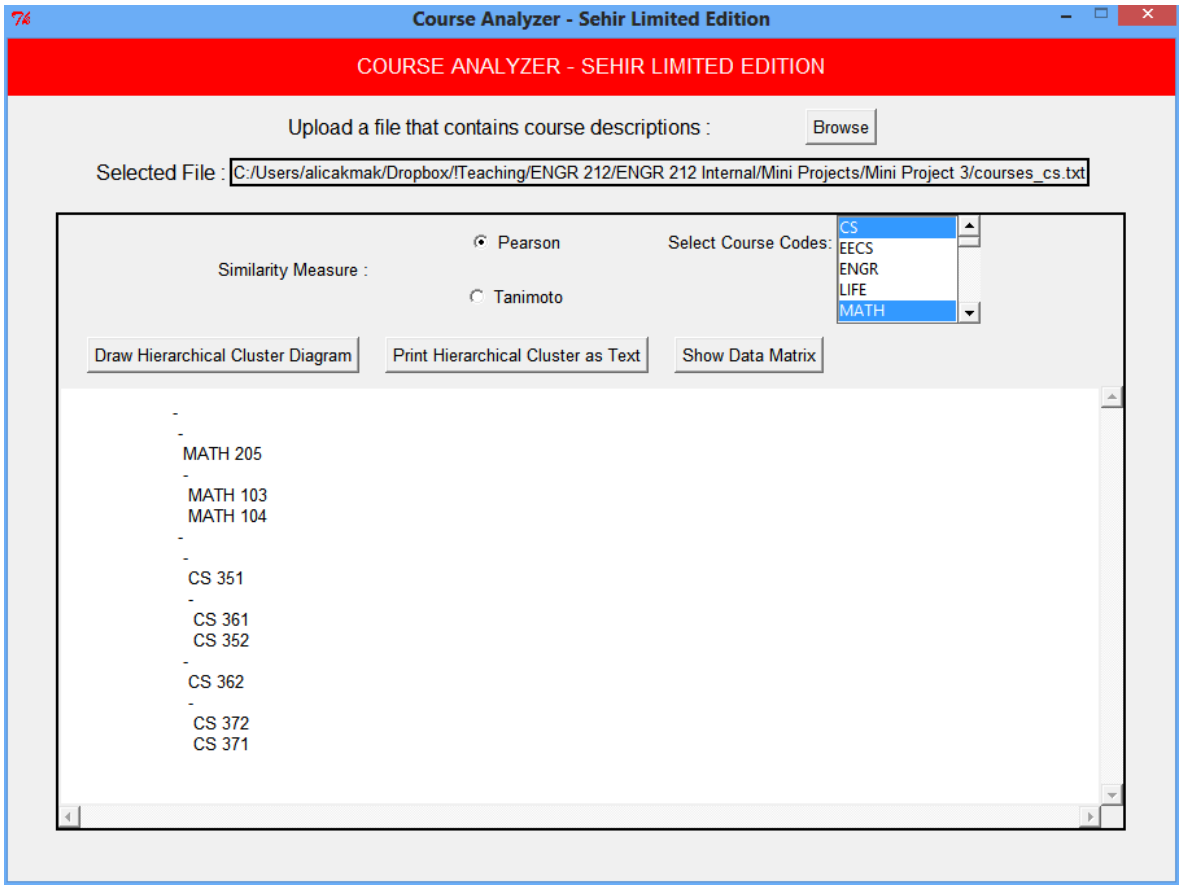

Figure 3

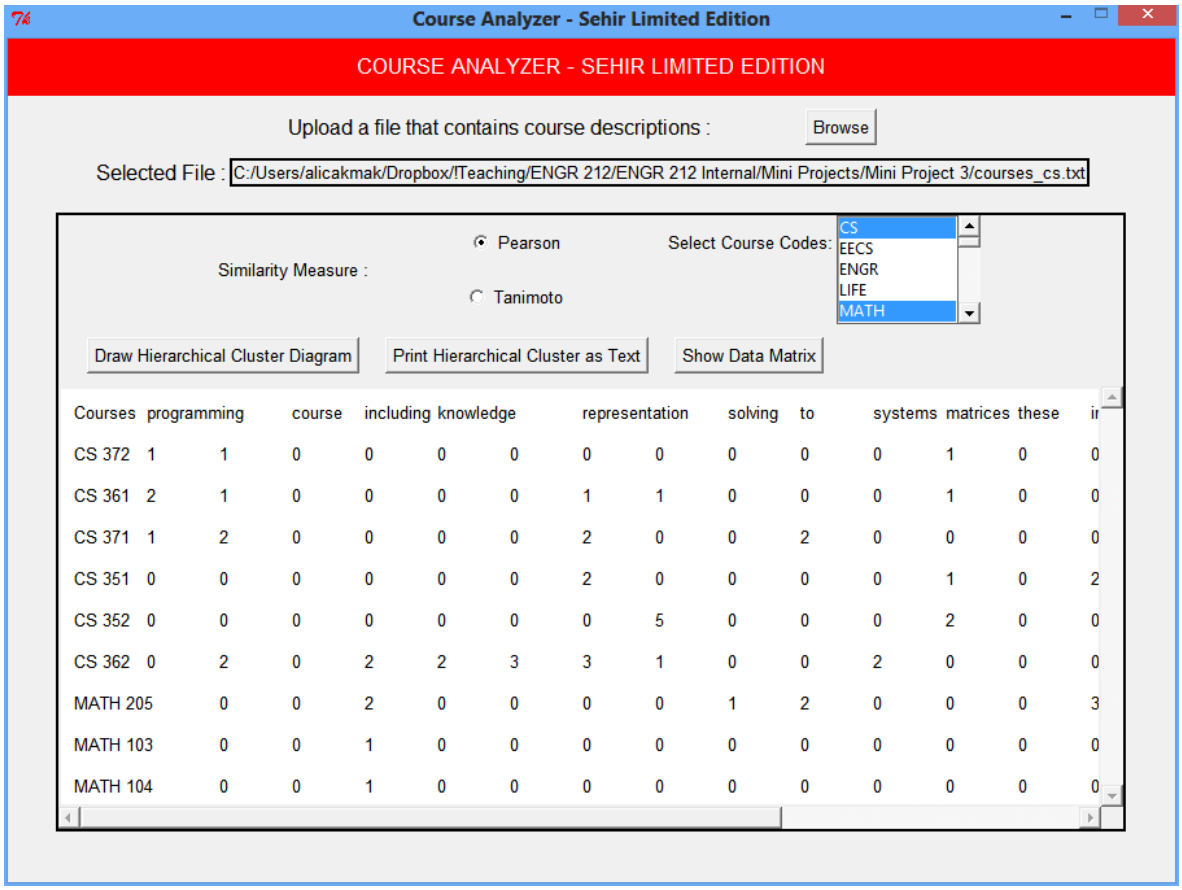

Figure 4

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

- For canvas scrollbar example, you may see the following piece: http://stackoverflow.com/questions/7727804/python-and-tkinter-using-scrollbars-on-a-canvas/7734187#7734187

- You may want to study drawdendogram function in clusters.py, and understand how it works. You may consider it to get inspired regarding how to produce similar drawing on canvas.

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 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 details).

- Submit your own code in a **single** Python file (Do **not** include clusterys.py that you import). 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.
  - Those who **do not** follow the above naming conventions **will get -5 off** of their project grade.

- Submit it online on LMS (Go to the Assignments Tab) by **17:00 on April 12, 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? :

Code Organization			Functionality					
Meaningful variable names (%3)	Classes and objects used (%4)	Sufficient commenting (%4)	Compiles? (20)	GUI Design (10)	Reading course descriptions / Populating course codes (10)	Drawing hiearhical clustering properly with scrollbars placed (40)	Printing hiearhical clustering as text properly (5)	Printing data matrix on canvas properly (5)

- 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. **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!**