CSCE 314 – Java Project – Group Creator

Objective:

Use these Java topics ***mostly*** covered in class:

[Abstraction](http://faculty.cse.tamu.edu/slupoli/notes/Java/PolymorphismAbstractionInterfacesNotes.docx)

[File I/O](http://faculty.cse.tamu.edu/slupoli/notes/Java/FileIONotes.docx)

[Collections](http://faculty.cse.tamu.edu/slupoli/notes/Java/CollectionsNotes.docx)

[Generics](http://faculty.cse.tamu.edu/slupoli/notes/Java/GenericsNotes.docx)

[Try/Catches](http://faculty.cse.tamu.edu/slupoli/notes/Java/ValidationNotes.docx)

[Comparators](http://faculty.cse.tamu.edu/slupoli/notes/Java/CollectionsNotes.docx)

[ENUM](http://tutorials.jenkov.com/java/enums.html)

**Part 2**

GUI in Java – I ([Word](http://faculty.cse.tamu.edu/slupoli/notes/Java/SWING_GUI_I_Notes.docx)) ([PlayList](https://www.youtube.com/playlist?list=PLC7fNkE1QplbO6KKYn_FhOr7bE0HuKg8J)) II ([Word](http://faculty.cse.tamu.edu/slupoli/notes/Java/SWING_GUI_II_Notes.doc)) ([PlayList](https://www.youtube.com/playlist?list=PLC7fNkE1QplYQTAr5ztM1tE7vQ9Hwdzhf))

III ([Word](http://faculty.cse.tamu.edu/slupoli/notes/Java/SWING_GUI_III_Notes.docx)) (\*May not be needed) IV ([Word](http://faculty.cse.tamu.edu/slupoli/notes/Java/SWING_GUI_IV_Notes.doc)) (\*May not be needed)

Application:

While mostly covered in the video linked [here](https://youtu.be/onqfm9cPBtw), you will create groups of 2 people based on responses (formula described in the video) from a survey in CSV (comma) form linked here.

Coding Standards:

There will be many things NOT in the UML below in order to keep it simple.

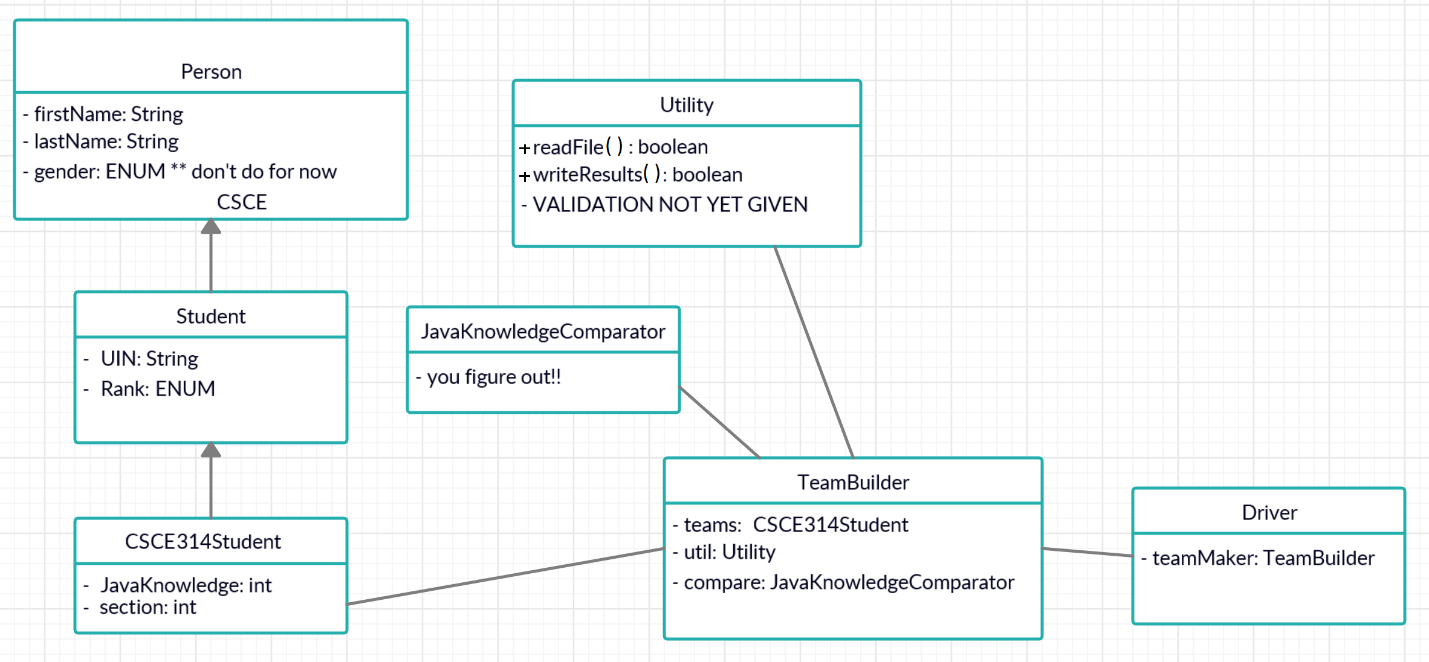
1. Setters/Getters/toStrings (“Complete class profiles”) for EACH class given below, ***except Utility, Driver, JavaKnowledgeComparator and Rank***.
2. Each file has a (header comments)
   1. description of what the class is doing as a part of the project
   2. Your gradebook name
   3. Your UIN
   4. Your email
   5. ***IN THAT ORDER***
3. Comments
   1. Other than the file header comments, you need to comment ***about*** 30% of your code
   2. ***prime times to comment your code***, adding, deleting, if/else statement, validating, etc… anything that is not 201 stuff.
   3. ***I will not give every example when to comment, if in doubt, COMMENT***
4. Any errors caught…will be logged in ErrorLog.txt
   1. the error description must state where (file/class) the error was caught and why
      1. think of this as debugging

Some assumptions:

1. Names (first and last) will always be separated by a space
2. The UIN is denoted by the first part of their email. [***87123AA3***@tamu.edu](mailto:87123AA3@tamu.edu)
   1. so 87123AA3 is their UIN
3. In the results, there likely will be someone by themselves. They become a “sub” for anyone sick.
   1. only those in the SAME section can be partners
4. There **will be more than 2 sections** within the data file
   1. you should sort by section, then knowledge
5. CREATE THE CODE FROM SCRATCH. You are not to use other “pre-packaged” code other than what Java gives you naturally (all imports shown in notes)
6. The overall layout of the file will always remain the same
7. FOR NOW… gender is not considered in the project. Yet.
8. Person (class/file) **must be** abstract.
9. Student (class/file) **must inherit** from Person.
10. 314Student (class/file) **must inherit** from Student.
11. JavaKnowledgeComparator (class/file) **must be** simple and just be used as a comparator with not much utility other than sorting order.
12. Any Java Collection data structure can be used.
13. TeamBuilder (class/file) ***should be*** where **most of the application work is conducted**. It will pull from all the other classes EXCEPT Driver.
14. Driver should be minimum. Probably less than 20 lines.
15. ENUM for RANK will be “Freshman”, “Sophomore”, “Junior”, “Senior”. But from the data file, Freshman == 1, Sophomore == 2, etc…
16. Results of grouping should be placed (within the same directory as the driver) in **results.txt** in this form:
    1. Name #1 (first then last)
    2. (UIN)
    3. (Knowledge)
    4. “—"
    5. Name #2 (first then last)
    6. (UIN)
    7. (Knowledge)

502 - Bowen Lupoli (671165121) (4) -- Jamila Booth (723223828) (2)

Code Setup:



Above is NOT a perfect UML, but enough to get you by for now. Notice those with () are functions. The return types are after the name. Notice, boolean in most of these function is just to return if the function was able to complete successfully.

[(how to read a UML)](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/)

**Part 2**

Part 1 must work before you can start part 2. Part 2 is adding Java GIU SWING. While the code is up to you, the GUI must look like below.



Notice, you will now display your results BOTH in file form and in the GUI textboxes on the interface shown. The “File” menu item will open a file open box, and you will be able to access the file through that interface.

Submission:

Your submission deadline date is on our calendar, but it will be due on ECampus by 11:59pm. ALL FILES will be ZIPPED (not tarballed, not 7zipped, not whatever…) together. There is no late submission, and none will be accepted for any reason.