**CPSC 230: Computer Science I**

**Comprehensive Programming Assignment**

**Overview**

Write a module **Hogwarts.py**

You work in the registrar’s office at Hogwarts School of Witchcraft and Wizardry. And you have a file of all senior students with their name, house and courses registered (***student\_records.txt*** provided). You have been asked to utilize the file to perform the following 3 tasks:

1. create an interactive platform where students can type in their name and check which house they belong to and what classes they are taking.
2. generate a file (class roster) for each individual class with the names of all students enrolled in it. The class roster files should be named after the course. eg. Potions.txt
3. determine which students are eligible for the following 2 job postings, based on the course they are taking:
   1. Auror position: needs to be taking: DADA, Transfiguration and Potions
   2. Ministry position: needs to be taking: Transfiguration, Charms and Potions

Your module should implement the following functions:

1. **processed\_line:** will take in a line from the file and return a list with 3 strings [name, house, courses]

For example, the first line should return the following:

['Harry Potter', 'Gryffindor', 'DADA,Transfiguration,Potions,Flying']

1. **my\_search:** will take in a student name, a file and a choice; and returns a string. Based on the choice, the returned string will be either the student’s house or the classes the student is taking.

For example:

student name = Harry Potter

choice = classes

should return 'DADA,Transfiguration,Potions,Flying'

1. **courses\_dictionary:** will take in a file and returns a dictionary with the courses as keys and a list of corresponding students as values.

For example, below is a sample of the dictionary:

{Arithmancy: ['Hermione Granger', 'Percy Weasley'], DADA: ['Harry Potter', ....], ... }

1. **career\_check:** will take in 3 lists and returns a list of elements common to all 3 lists. Used for checking the eligibility of the student to apply for the career opportunity.

For example, for the Auror position, the 3 lists are students taking 1) DADA 2) Transfiguration 3) Potions.

Each list is list of students taking a course required for the position.

eg. DADA\_list = ['Harry Potter', 'Hermione Granger', 'Ron Weasley',...]

* Call this function again for a different job, with different course list.
* For initial testing: create your own 3 lists.

**Note:** the following two are procedures. They will be used for creating the different class rosters.

1. **write\_file:** will take in a list of names and a filename, and writes the names from the list into the file (one name per line).
2. **class\_roster:** will take in a dictionary and writes to multiple files the names of students in that class. Remember to name the files appropriately

*Hint:* Filenames should be dictionary keys. Names to be written are associated values.

Once you have the individual functions tested and validated, you can integrate them together to accomplish the 3 tasks mentioned above. Keep in mind that you should be calling functions within other functions when necessary.

* Your module should prompt the user for a name and output their house or classes.
* It should create one file per class with the enrolled student names.
* It should display a list of students eligible for each of the 2 job openings.

The file name to be read from will be provided by the user via an input. You should also incorporate error handling to ensure the file name inputted by the user is valid and keep prompting the user till they enter the correct filename.

**Due Date**

You may work in teams of 4-5 or as a table, only 1 submission per team is required. Please make sure to include all team members names in the README as well as in the source file. The README should **describe each team member’s contribution clearly**. Any additional features/modifications/known errors should also be mentioned.

This assignment is due at 11:59 pm on the specified due date. Submit via Canvas. It should be labeled firstinitiallastname\_CompAssignment based on the student submitting the files.

**Grading**

Your program will be evaluated for correctness and elegance. You should make sure your code is properly commented and obeys standard naming conventions.