Final Project - COP 3330- Spring 2023

_____*Instructions*_____

1) This project can be completed in groups (max 3 per group) but submission in Webcourses is individual. Don't copy anybody's work nor give your code to anybody unless you **seriously** worked together on the project.

2) Apply the Java naming conventions. See *JavaNamingConventions.pdf* posted on Webcourses (deduction up to 10 points if this not done properly)

3) If you work with a classmate, your .java file must contain the following comment:

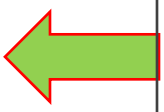
```
/*
- COP 3330 Final Project.
- Names (first and last names) of students who worked together on the project.
- (optional) Add anything that you would like the TA to be aware of
*/
```

Example:

```
/*
- COP 3330 Final Project
- Hatim Boustique, Ericka Edwards and Zayd Majdoubi
  (In this case, Hatim, Erika and Zayd must submit the same .java file on Webcourses)
*/
```

Your .java file should be similar to:

```
/*
-   COP 3330 Final Project
-   Hatim Boustique, Ericka Edwards and Zayd Majdoubi
*/


public class FinalProject {

    public static void main ( String[] arguments ){

            //Your test code goes here
    }

    . . .
}


class Abc {


    //code of class Abc
    . . .
}


class Xyz {


    //code of class Xyz
    . . .
}
```



FinalProject.java

Note well:
Students will not receive any credit if they don't submit the .java file by the deadline! Submissions by email will not be considered for a grade. Students must submit their projects on Webcourses by uploading the .java file. It is the responsibility of the students to check their submissions to make sure that the file they submitted is indeed the right file and it is a readable file!

_____*The Final Project statement*_____

This project is about implementing a scheduling system of lectures/Labs offered at the Computer Science Department here at UCF (CSD), along with enrolling students to the lectures/labs. More specifically, a class offered at the CSD is either a lecture (LEC) or a lab (LAB). For each class (lecture or lab), we store a Course Number or crn (a unique five-digit number assigned for each lecture/lab), a prefix (like COP3330), title (like Introduction to Object Oriented Programming), location (like CB 2-201), Modality (Online, Face-to-Face or Mixed Mode), and whether the lecture is a graduate or undergraduate. Note that online lectures don't have labs, and some mixed or face to face lectures require a lab. For example, the CSD offers COP 3223/Introduction to Programming with C require labs. The list of lectures and labs is provided in lec.txt, a text file that provided with this project.

Lectures are taught by Faculty only, and Labs are taught by Teaching Assistants (TAs) only. *Faculty* are characterized by a name, UCF id number (a 7-digit number), rank (professor, associate professor, assistant professor or adjunct) and a list of lectures taught by that faculty. *TAs* have a name, UCF id number, list of Labs supervised, Advisor (a faculty who may not be teaching anything), expected degree (MS or PhD) and a possible list of lectures and labs taken by the TA (TAs are students themselves!). A *Student* is characterized by a name, UCF id number, type of student as either graduate or undergraduate, and a possible list of lectures and Labs taken.

Your code is expected to handle as many Faculty, TAs and Students as needed. *Your project is expected to have ONE ArrayList list (or any other data structure of your choosing) to store the Faculty, TA and Students objects (up to 30 points deduction if faculty, TA and Student objects are stored in separate data structures).* Now, where and how to deal with the lectures offered is up to you; you may work on the offered lectures and labs using any data structure you choose.

It is your responsibility to create an inheritance hierarchy of classes to demonstrate the use of code reusability. Bottom Line: Your project should not contain redundant code. Furthermore, any class from which you are not instantiating objects must be made

abstract (10 points deduction here). The use of interfaces is left to you to decide on, but that is not a requirement.

You are asked to add the necessary code to prevent the code from crashing. For example, when asked to enter how many lectures to assign to a faculty, and if the user enter a non integer value, then your code should catch the thrown exception to prevent the code from crashing. In this project, we first ask the user to enter the absolute path of the file that contains the list of lectures/labs offered.
Once again, your code should not crash!

In addition, create a class that you must call *IdException* so that whenever the user enters the wrong format when prompted to enter the UCF id, an object of the IdException class is thrown and caught to display a message saying:

```
>>>>>>>>>>>Sorry incorrect format. (Ids are 7 digits)
```

The list of lectures and associated labs is provided in lec.txt, a text file whose structure can't be modified (Your code can't just run for the provided lec.txt content but the structure is to remain the same. See below).

Lines in *lec.txt* are of the form:

CRN,PREFIX,LECTURE TITLE,GRADUTE/UNDERGRADUTE/MODALITY,BUILDING CODE-ROOM NUMBER,LAB(YES/NO)

Examples:

89745,COT6578,Advanced Computer theory,Graduate,F2F,PSY-108,No

32658,COT6578,Advanced Computer theory,Graduate,Mixed,LPS-35,No

That means that the CSD offers two sections of COT6578, one is F2F and the other is online. Both of those sections don't have labs.

When a lecture has a lab, the labs' information is provided right after the lecture information

Example:

69745,COP5698, Programming Languages,Graduate,F2F,CB2-122,YES

19745,MSB-123

36598,PSY-100

20315,HSA1-116

That means the COP5698 with crn 69745 has three labs. 19745, 36598 and 20315 are the crns of the labs. MSB-123, PSY-100 and HSA1-116 are the building-room numbers where those labs are scheduled to take place.

Note that if the Modality is Online, then the line of the lec.txt has the form:

CRN,PREFIX,LECTURE TITLE,GRADUTE/UNDERGRADUTE/ONLINE

Example:

89745,COP4578,Software Engineering,Undergraduate,Online

Your code is expected to implement a menu-based application that allows the user to add a new faculty, a new student and a new TA when a lab is required. See below for more details…

The options to implement are:

1- Add a new faculty to the schedule.

(This requires assigning TAs to the labs when applicable)

2- Enroll a student to a lecture (and to one of its labs if applicable)

3- Print the Schedule of a faculty.

4- Print the schedule of an TA.

5- Print the schedule of a student.

6- Delete a scheduled lecture

(Deleting a lecture requires deleting its labs and updating any student's schedule accordingly)

7- Exit Program

Your code must check the following:

- A lecture can't be assigned to two faculty and a Lab can't assigned to two TAs, and an TA can do more than one lab.
- A student can't take two lectures with the same prefix.

- UCF ids are unique: When entering the id of any person, do a search to check that the id isn't already used.
- A lab can't be assigned to two lectures.

Here are the details of each of the menu options:

1- Add a new faculty to the schedule.

Your code collects the UCF id, name, rank, office location (no required form for the office location-any string would do it!) and how many lectures to assign to this faculty. Recall the crns are unique (each lecture/lab has a unique crn).

Enter UCF id:----

If the id exists, then skip name, rank and office location.

Enter name:----

Enter rank:----

Enter office location:----

Enter how many lectures:

Enter the crns of the lectures to assign to this faculty

If a lecture has labs, ask to enter the UCF id of the TA for each lab (a TA may do more than one Lab). This may require entering a new TA to the system, and in this case, you need to ask for the remaining information of the TA. Keep in mind that a TA can be a student.

2- Enroll a student to a lecture (and to of of its labs if applicable)

Enter UCF id:----

Enter name:----

Enter the crns of the lectures (zero is an acceptable entry here)

If a lecture requires a lab, randomly pick a lab for the student (no caps on how many students to enroll in a lab) (Use the built in Java random generation of a number)

3- Print the Schedule of a faculty.

Enter UCF id of the faculty:-------

Then Print the UCF id, name and the crns of the lectures taught by the faculty (No need to display anything else).

4- Print the schedule of an TA.

5- Print the schedule of a student.

6- Delete a scheduled lecture

(Deleting a lecture requires deleting its labs as well and updating any faculty/student record who is teaching/taking that lecture)

7- Exit Program

If a lecture has been deleted update the file lec.txt.

What matters the most when your project is graded:

1) No redundant code (Use inheritance), and try not to write too much code; use the Java built-in classes whenever that is applicable. Up to 30-point deduction
2) When your project is graded, the lectures and labs provided in lec.txt may change but the file structure stays the same.
3) UCF ids must be 7 digits, and are unique. 20-point deduction if that is not checked
4) No IdException (throwable) class in your code: 10-point deduction
5) The use of interfaces is not required but permissible.
6) You may use other data structures to work on the lectures/labs. One idea is to load the offered lectures/labs from lec.txt into Arraylist(s). But don't forget to update lec.txt when a lecture is deleted! 10-point deduction if lec.txt isn't updated
7) Program crashes (for whatever reason). 20-point deduction
8) All the faculty, TA and student objects are to be stored in ONE data structure (one array, one ArrayList,…etc). Up to 30-point deduction
9) The provided sample run should give you a clear idea on how your code should run. Note well that the spacing and the order in which things are tested in the provided sample run doesn't matter. That is: all your menu options should all run properly. 10-point deduction if any menu option that is not done properly

----- SAMPLE RUN ------

Enter the absolute path of the file: **c:\abc\xyz\lec**

Sorry no such file.

Try again: **c:\abc\xyz\lec.txte**

Sorry no such file.

Try again: **c:\abc\xyz\lec.txt**

(No need to check if it is the file that has the lectures)

File Found! Let's proceed…

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: 1

Enter UCF id: **1570881** (ids are to be 7 digits)

Enter name: **Kyle Johnson**

Enter rank: **Professor** (professor, PROfeSsor are acceptable)

Enter office location: **LRT-125** (no required format for office location)

Enter how many lectures: **2**

Enter the crns of the lectures: **69745 66636**

(Assume the crns entered are correct-but check for already assigned crn)

[66636/DIG2158/Introduction to Digital Systems] Added!

[69745/COP5698/Programming Languages] has these labs:

19745,MSB-123

36598,PSY-100

20315,HSA1-116

Enter the TA's id for 19745: **1570884**

(A student can't be a TA for a lecture in which that student is taking)

TA found as a student: Emma Jones

(This means that Emma isn't a student in Programming Languages)

TA's supervisor's name: **Douglas Richardson**

Degree Seeking: **PhD**

Enter the TA's id for 36598: **1570802**

Name of TA: **Erick Johann**

TA's supervisor's name: **Tamra Singh**

Degree Seeking: **MS**

Enter the TA's id for 20315: **1570884**

(note that this TA. Emma is now assigned 2 labs of Programming Languages)

[69745/COP5698/Programming Languages] Added!

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **2**

Enter UCF id: **1570802**

Record found/Name: Erick Johann

Which lecture to enroll [Erick Johann] in? **60045**

(Make sure that Erick isn't assigned as TA for 60045)

[COP5690/Programming Languages II] has these labs:

19005,MSB-103

30008,PSY-107

20300,HSA1-16

[Erick Johann] is added to lab : **30008** (30008 is picked randomly)

Student enrolled!

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **4**

Enter the UCF id: **1597538**

Sorry no TA found.

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **5**

Enter the UCF id: **1570802**

Record Found:

Erick Johann

Enrolled in the following lectures

[COP5690/Programming Languages II]/[Lab: 30008]

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: 3

Enter the UCF id: **1570881**

Kyle Johnson is teaching the following lectures:

[DIG2158/Introduction to Digital Systems][Online]

[69745/COP5698/Programming Languages] with Labs:

[19745/MSB-123]

[36598/PSY-100]

[20315/HSA1-116]

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **6**

Enter the crn of the lecture to delete: **36637**

[36637/SOF2058/Introduction to Software] Deleted

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **6**

Enter the crn of the lecture to delete: **69745**

[69745/COP5698/Programming Languages] Deleted

(69745 and its labs are deleted entirely from the system. When a lecture is deleted, it is deleted from any student and faculty's schedule, and its Labs are deleted from any TA's schedule.)

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **4**

Enter the TA's UCF id: **1570884**

No TA with this id

(Note that when we deleted lecture **69745**, the associated labs are all gone, and consequently, Emma Jones is no longer hired)

****

Choose one of these options:

1- Add a new Faculty to the schedule
2- Enroll a Student to a Lecture
3- Print the schedule of a Faculty
4- Print the schedule of an TA
5- Print the schedule of a Student
6- Delete a Lecture
7- Exit

Enter your choice: **7**

You have made a deletion of at least one lecture. Would you like to print the copy of lec.txt? Enter y/Y for Yes or n/N for No: **t**

Is that a yes or no? Enter y/Y for Yes or n/N for No:**n**

(Note that lec.txt must be updated whenever a deletion of a lecture occurred!)

Bye!