COMP 2215: HOMEWORK – 1

Implementing A simple Student Course Registration System

You are tasked with implementing a simple course registration system. Your system is required to include 6 classes in total. These are:

1. Course.java
2. CourseRegistration.java
3. Instructor.java
4. Student.java
5. User.java
6. Main.java

Make sure to add all required getters / setters and constructors as you see fit for each of these classes. Every class also required to have a toString method that prints all the attributes of an object in an orderly manner.

Course.java: A class that is going to be used to represent a course within the system. Will have following attributes

- CourseID: Every course is required to have this field as in identifier. Courses ought to start from ID 1 and increase as the new courses are added.
- Course Code: A String that is used as an abbreviation for the course. For example, COMP2215 is the course code for this class.
- Course Name
- ECTS

User.java: A class to represent common attributes between a Student User of the system as well as an Instructor user of the system. These are mainly:

- UserID: Every user of the system is required to have this field as in identifier. users ought to start from ID 1 and increase as the new users are added.
- Name
- Surname
- birthdate (dd/mm/yy) format
- gender
- phone Number
- login Email
- password

This class also should contain an abstract method called listAllCourses that will do different things depending on the subclass. Make sure to make this class non-instantiable. Meaning a user within the system must be either student or an instructor.

```
public abstract void listAllCourses(List<CourseRegistration> registrations);
```

Student.java: Student is a **subtype of a user** with the following unique fields:

- StudentID : Should be a different identifier aside from the UserID we've defined.
- Years of Study

Student class is required to have following methods:

```
public int calculateTotalECTS(List<CourseRegistration> registrations){...13 lines }

@Override
public void listAllCourses(List<CourseRegistration> registrations) {...13 lines }
```

CalculateTotalECTS method is supposed to go through and filter all the registration records and filter those records based on student ID. This method will simply calculate total credits (ECTS) of all the classes a particular student registered.

ListAllCourses for Student objects should find all the classes a particular student takes and prints them from 1 to n. n being the number of classes that student takes.

A sample screenshot can be seen below.

```
----------------------------------------
Student :Student ID:20  Name :Sonia Wiley       BirthDate:10/07/2001
Courses Taken:
1)CE 1200 Data Structures
2)CE 1600 Mobile Software Development
Total ECTS :11
----------------------------------------
Student :Student ID:21  Name :Sasha Romero      BirthDate:23/06/1995
Courses Taken:
1)CE 1100 Introduction to Programming
Total ECTS :5
----------------------------------------
Student :Student ID:22  Name :Vivian Mclaughlin BirthDate:25/05/1997
Courses Taken:
1)CE 1300 Database Systems
Total ECTS :6
----------------------------------------
Student :Student ID:23  Name :Baylee Stevenson  BirthDate:15/11/1997
Courses Taken:
1)CE 1300 Database Systems
2)CE 1500 Web Development
Total ECTS :9
----------------------------------------
```


Instructor.java: is a **subtype of a user** with the following unique fields:

- Office Phone
- Employment Type (Professor or Ast. Prof)
- List of Taught Courses (Course ArrayList)

Instructor class is required to have following methods:

```
@Override
public void listAllCourses(List<CourseRegistration> registrations) {...33 lines }
```

listAllCourses will first list all the courses taught by a particular instructor and then in the second step it should print out an attendance list (A list of students registered to that class) for every class taught by that particular class. Sample output can be seen below.

```
Professor:Instructor Type:Professor     Name :Derrick Mcguire   BirthDate:12/09/1981    Gender:Male     Phone Number:(452) 449-4697
Taught Courses:
CE 1200 Data Structures
CE 1500 Web Development
CE 1600 Mobile Software Development

Attendance List for Class Data Structures
1)Student ID:10 Irvin Rhodes
2)Student ID:13 Elyse Mcpherson
3)Student ID:15 Karlee Hester
4)Student ID:20 Sonia Wiley
Attendance List for Class Mobile Software Development
1)Student ID:7 Ernesto Ware
2)Student ID:18 Ashley Ryan
3)Student ID:20 Sonia Wiley
Attendance List for Class Web Development
1)Student ID:3 Cordell Monroe
2)Student ID:7 Ernesto Ware
3)Student ID:8 Reilly Morrow
4)Student ID:9 Caiden Gregory
5)Student ID:11 Alberto Castaneda
6)Student ID:19 Alivia Morse
7)Student ID:23 Baylee Stevenson
------------------------------------------------
```

CourseRegistration.java: A container class that will map a particular student to a particular course thereby defining a course registration. This will have following fields and a constructor.

```
public class CourseRegistration {
    private Student student;
    private Course course;
    private String registration_date;
    private String registration_time;

    public CourseRegistration(Student student, Course course, String registration_date, String registration_time) {
        this.student = student;
        this.course = course;
        this.registration_date = registration_date;
        this.registration_time = registration_time;
    }
```

Main.java:

1. Create 25 Student objects / 3 Instructor Objects / 6 Course Objects.
2. Randomly assign every course to an instructor.
3. Randomly assign 1 or 2 (Randomize as well) courses to Every student. <u>You can for now ignore the situation a student may end up taking same course twice as solving this will require more effort than this assignment needs.</u>

4. Put Every User into a User Arraylist and Print out **ListAllCourses** and calculateTotalECTS for Students and ListAllCourses for every instructor. Use downcasting objects whenever necessary. You'll get an output like below:

```
---------------------------------------------
Professor:Instructor Type:Ast. Professor        Name :Whitney Reid      BirthDate:01/02
Taught Courses:
CE 1300 Database Systems

Attendance List for Class Database Systems
1)Student ID:2 Brett Frey
2)Student ID:4 Ezra Delgado
3)Student ID:5 Marcus Guzman
4)Student ID:14 Shyanne Woodard
5)Student ID:15 Karlee Hester
6)Student ID:22 Vivian Mclaughlin
7)Student ID:23 Baylee Stevenson
8)Student ID:24 Alannah Clark
---------------------------------------------
Student :Student ID:1   Name :Tristian Wilkins  BirthDate:11/08/1995    Gender:Male
Courses Taken:
1)CE 1400 Software Engineering
Total ECTS :4
-----------------------------------------
Student :Student ID:2   Name :Brett Frey        BirthDate:31/12/1995    Gender:Male
Courses Taken:
1)CE 1100 Introduction to Programming
2)CE 1300 Database Systems
Total ECTS :11
-----------------------------------------
Student :Student ID:3   Name :Cordell Monroe    BirthDate:22/01/2002    Gender:Male
Courses Taken:
1)CE 1500 Web Development
2)CE 1100 Introduction to Programming
Total ECTS :8
```

You can find the sample data for your task in the supplied *.txt files or you can create your own.

Submit and upload:

1. *.java files
2. A screenshot showing your program output with taskbar showing date and time.

Good Luck