ASSIGNMENT

Create the program by java. It includes

1. Person class, it contains variable data : code, name , address and some variable methods : constructors, getter, setter and overriding the method toString() to return formatted string: code-name-address And some class variable TAX = 10%
2. Student class extends Person class and implement Interface Comparable and add some data:

- Double grade
- Date valid

And some methods:

+ Constructors
+ getter, setter
+ override the method toString() to return formatted string : code-name-address-grade
+ override the method compareTo(): compare the grade between two student
+ isValid: check if student has valid date is more than 6 years to current date

3. Professor class extends Person class and add

- Enum "PositionEnum": PROFESSOR, ASSOCIATE PROFESSOR, HONOR PROFESSOR
- Enum "EducationLevel": BACHELOR, MASTER, DOCTOR
- some data: int experience (>0), int basic salary (default is 1000) , PositionEnum postion, EducationLevel education

List<Person> arr , int count (list of students)

- and some methods: constructors, getter, setter and override the method toString to return formatted string: code-name-address-experience-real salary
- Some class variable and method: COF = 0.33 and STEP = 3 and SUPPOS = {if ASSOCIATE PROFESSOR = 0.5, PROFESSOR = 1.25, HONOR PROFESSOR = 0.75} and SUPLEV = {if MASTER = 0.1 and DOCTOR = 0.25}
- **And some methods:**

+ Real salary = basic salary * (experience / STEP) * (1+COF)*(1+SUPPOS)*(1+SUPLEV)

+ Calculate the annual income = real salary * 12 * (1-TAX)

+ bool Add(Student x): to add student X into array and return true if add successfully
+ AddAllStudent: read all the student data from file with format
+ Remove the student with code
+ Update the information student

+ RemoveInvalid: remove all student has valid date is more than 6 years to current date
+ Student find (findcode) : return student that the student's code=findcode
+displayStudentsAll(): display list of students
+ Sorting the list of student in descending
+ Get all student has the most grade
+ Get all student has at least grade
+ Find all distribution of grade
+ Find an average of grade
+ Find distinct grade
+ Find all student has the same name
+ Find all distinct name in class
+ OutAllStudent: write all student data into file with format

4. **You must think 10 distinct methods and add them into class Professor**
5. A class called Demo. It contains main method:
   In main method:
   You create **an object professor** and then a menu will show:
   1. AddAll students to professor's list of students
   2. Display the professor's list of students
   3. Find a student
   4. …….
   5. exit