CS 210 –SPRING QUARTER 2025 - TAESIK KIM – TAESIK.KIM@BELLEVUECOLLEGE.EDU

– ASSIGNMENT 3 –

CLASSES + ARRAYLISTS + SCANNER + CONSTRUCTOR + OVERLOADING

DUE: **Tuesday**, **06/17/2025**, 11:59 P.M.

" Do it yourself "
- Taesik Kim, Ph.D.

I) (40 pts.) - This assignment revisits the student parsing program from previous practices, but challenges you to restructure the component pieces of the program to create a cleaner, more succinct Main(). You will generate a People class of object and load an ArrayList with person objects, then report the contents of that ArrayList. To do so, you must perform the following:

A) (10/40 pts.) - Create a class file "People.java" (which will generate People.class upon compile). People.java will have four methods and one constructor to

   1) read a .txt file '`people.txt`'
   2) generate:
      - List of all students OR all teachers
      - List of female or male students OR teachers whose age is greater than or equal to given age
      - List of female or male students OR teachers whose age is greater than or equal to given age1 and less than or equal to given age2
      - List of students OR teachers whose age is greater than or equal to given age

B) (5/40 pts.) – When you instantiate your class with new, you must use the text file name (eg. People p = new People ("people.txt"))

C) (10/40 pts.) – You are required to use "Constructor", "ArrayList" and "Overloading" concepts.

D) (5/40 pts.) Once your `People.java` is functional, generate a file "`Assg03_yourName.java`". `Assg03_yourName.java` should contain only one `Main()` method.

   [Example of Main() in `Assg03_yourName.java`]
   Note:
   - **Use the following code to check if your program is working properly. You must test all the examples shown bellow.**

```
public static void main(String[] args)throws FileNotFoundException  {
  People p =new People("people.txt");
  p.list("s"); // all students
  p.list("t"); // all teachers
  p.list("s","f", 18);  // female students age>=18
  p.list("s","m", 20);  // male students age>=20
  p.list("t","f", 30);  // female teachers age>=30
  p.list("t","m", 35);  // male teachers age>=35
```

```
        p.list("s","f", 15, 18); //female students age>=15 & age<=18
        p.list("s","m", 15, 20); //male students age>=15 & age<=20
        p.list("t","f", 30, 40); //female teachers age>=30 & age<=40
        p.list("t","m", 30, 40);  //male teachers age>=30 and age<=40
    }
```

E) (10/40 pts.) Upload three files:

- "Assignment3_yourName.java" to Canvas.
- `People.java` to Canvas
- Capture all output, paste it into a PDF file, and upload it.

F) If you used someone else's code (part or full), you will get -120 BONUS points. In some cases, you will be asked to explain to me how your program does work.

G) (-20 pts.) - Include appropriate program documentation and formatting including:

1) Your first and last name
2) Your contact information
3) Your student ID number
4) The date
5) A short description of the program's function
6) Comments necessary to explain the operation of your program
7) Proper indentation

II) A Sample log of execution is as follow :

**Note: No user inputs are a part of this assignment*

//Test

```
p.list("s");              // all students
p.list("t");              // all teachers
p.list("s","f", 18);      // female students age>=18
p.list("s","m", 20);      // male students age>=20
p.list("t","f", 30);      // female teachers age>=30
p.list("t","m", 35);      // male teachers age>=35
p.list("s","f", 15, 18); //female students age>=15 & age<=18
p.list("s","m", 15, 20); //male students age>=15 & age<=20
p.list("t","f", 30, 40); //female teachers age>=30 & age<=40
p.list("t","m", 30, 40); //male teachers age>=30 and age<=40
```

Note:

When printing the type, it must be either teacher or student, not t or s.

When printing the gender, it must be either female or male, not f or m.

// output

```
List of people
type: student
John
Jack
Chloe
Racheal
```

Olivia
Mia
Malja

List of people
type: teacher
.
.
.

List of people
type: student gender: female age: >= 20
.
.
.

.
.