

COP 3330- *Fall 2023*-Final Project

- 1) Projects can be completed in groups (max 4 per group) but submission onto WebCourses is individual. Once again, don't copy anybody's work nor give your code to anybody unless you **seriously** worked together on the project.
- 2) Please Submit one .java file. It should look like this:

```
public class FinalProject {  
  
    public static void main(String[] args){  
        //Test code goes here  
    }  
}
```

```
//-----  
class Abc{  
  
}  
  
//-----
```

```
class Xyz{  
}  
}
```

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

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

Example:

```
/*  
- Final Project  
- Hatim Boustique, Ericka Edwards and Jamal Dubois  
*/  
  
public class FinalProject {  
  
    public static void main(String[] args) {  
        //Test code goes here  
    }  
}
```

```
    }
}
//-----
interface Int1 {
}

//-----
class Abc{
}

//-----
class Xyz{
}
```

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 readable .java file!)

Final Project Statement

Please read this entire statement carefully before you start doing anything...

This project involves implementing a simple university personnel management program. The program contains three kinds of objects: Staff, Student, and Faculty. For each object, the program stores relevant information such as university ID, name, etc. Different information is stored depending on the type of object. For example, a

student has a ***gpa***, a faculty has a ***title*** and ***department*** (professor, mathematics).

For each of any class data member, your program must include the **getters** and the **setters**, and each class must include at least **two constructors**. The main goal of this Project is to demonstrate the use of inheritance, abstract classes, abstract methods, and method overriding.

The ***Student*** class has the ***private*** attributes:

- full name : String
- id: String
- gpa: double
- Number of credit hours currently taken: int

For a faculty, we need a:

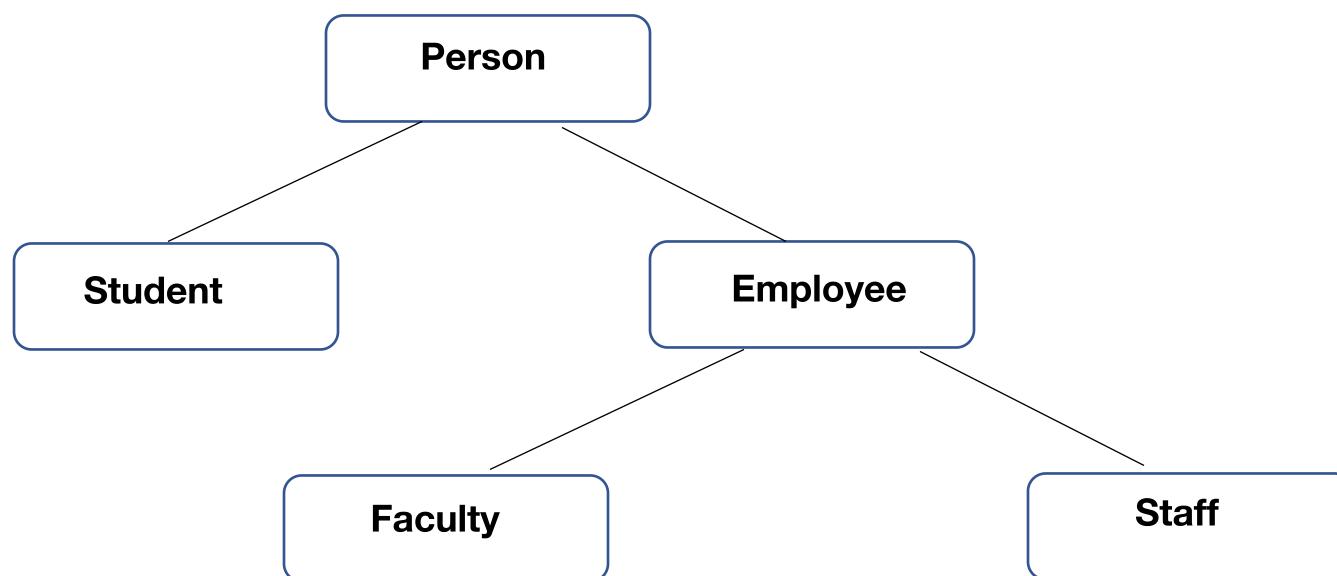
- full name: String
- id: String
- department (mathematics, engineering or english): String
- Rank (professor or adjunct): String

For a staff, we need a:

- full name: String
- id: String
- department (mathematics, engineering or english): String
- status (part time or full time): String

Students in this college pay \$236.45 per credit hour in addition to a \$52 administrative fee. Your code should generate a tuition invoice (**a method within the class Student**). Note that students get 25% off total payment if their **gpa** is greater or equal to 3.85.

Your code should implement the following inheritance hierarchy:



Both classes ***Student*** and ***Employee*** inherit from the *abstract* class ***Person***. The abstract class Person has what is common to a ***Student*** and an ***Employee (Faculty or Staff)***. The class ***Person*** must include the signature:
public abstract void print();

The abstract method ***print*** is overridden to print the fee invoice for a student and to print the information for a faculty or a staff member. It is left to you (the programmer) to come up with other abstract methods if you see fit(this is optional).

The class ***Employee*** should also be abstract, and it is supposed to include what is common to a ***staff*** and a ***faculty***.

Test your code with ONE array of size 100 of type Person. You may choose to use any data structure of your choosing such as an ***ArrayList***, ***Linked List***,...etc. The bottom line is that all objects of type ***Student***, ***Faculty*** and ***Staff*** are stored within the same data structure.

The sample run below should give you a clear idea about how your code should run. The user's entry is marked in bold so you can tell what your code should display to the screen and what the user enters.

Please note well that:

- 1) Your code should run as shown on the sample run below (However, the TA will not deduct points because you skipped two lines instead of three or your tuition invoice has 56 hyphens instead of 63!).
- 2) When asked to enter the faculty's department, **matheMatics** and **MathematiCs** are considered to be the same, and your program should display **Mathematics** if faculty information is to be displayed to the screen. However, if the user enters **Mathematics department**, then this is an invalid entry. Consider these

departments only: Mathematics, Engineering and English. As for the rank of a faculty, consider these ranks only: Professor and Adjunct.

- 3) Your code should handle all exceptions.
- 4) Note that any person's id must be of the form ***LetterLetterDigitDigitDigitDigit*** like ***ek7894***. See sample run below. We require that ids are checked for duplication. That is: No two persons can have the same id!
- 5) When option 8 is chosen to exist the program, the user is presented with the option to create a report that has the most up-to-date data from the array (or ArrayList list...) saved into a text file (report.txt). A sample report is provided at the end of this document.
- 6) The final report should print the students sorted by descending gpas or descending names. For how to sort the students (by gpa or name), you may implement one of the Java's interfaces Comparable or Comparator. Where to do that in your code is something that you need to figure out.

Sample Run: (Below is how your code should run)

```
Welcome to my Personal Management Program
```

```
Choose one of the options:
```

- 1- Enter the information a faculty
- 2- Enter the information of a student
- 3- Print tuition invoice for a student
- 4- Print faculty information
- 5- Enter the information of a staff member
- 6- Print the information of a staff member
- 7- Delete a person
- 8- Exit Program

Enter your selection: **2**

Enter the student info:

Name of Student: **Julia Alvarez**

ID: **j1254**

Invalid ID format. Must be LetterLetterDigitDigitDigit

ID: **ju1254**

Gpa: **3.26**

Credit hours: **7**

Student added!

- 1- Enter the information a faculty
- 2- Enter the information of a student
- 3- Print tuition invoice for a student
- 4- Print faculty information
- 5- Enter the information of a staff member
- 6- Print the information of a staff member
- 7- Delete a person
- 8- Exit Program

Enter your selection: **2**

Enter the student info:

Name of Student: **Matt Jones**

ID: **ma0258**

Gpa: **2.78**

Credit hours: **0**

Student added!

- 1- Enter the information of the faculty
- 2- Enter the information of the two students
- 3- Print tuition invoice
- 4- Print faculty information
- 5- Enter the information of the staff member
- 6- Print the information of the staff member
- 7- Delete a person
- 8- Exit Program

Enter your selection: **A**

Invalid entry- please try again

- 1. Enter the information of a faculty
- 2. Enter the information of a students
- 3. Print tuition invoice
- 4. Print faculty information
- 5. Enter the information of a staff member
- 6. Print the information of a staff member
- 7. Delete a person
- 8. Exit Program

Enter your selection: **1**

Enter the faculty info:

Name of the faculty: **John Miller**

ID: **jo7894**

Rank: Instructor

"Instructor" is invalid

Rank: Assistant Professor

"Assistant Professor" is invalid

Rank: Professor

Department: Engineering

Faculty added!

1. Enter the information of a faculty
2. Enter the information of a students
3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **3**

Enter the student's is: **ju1254**

Here is the tuition invoice for Julia Alvarez:

Julia Alvarez

ju1254

Credit Hours: 7 (\$236.45/credit hour)

Fees: \$52

Total payment (after discount): \$1,707.15

(\$0 discount applied)

1. Enter the information of a faculty
2. Enter information of a students
3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **3**

Enter the student's is: **ja1954**

No student matched!

1. Enter the information of the faculty
2. Enter information of the two students

3. Print tuition invoice
4. Print faculty information
5. Enter the information of the staff member
6. Print the information of the staff member
7. Delete a person
8. Exit Program

Enter your selection: **4**

Enter the Faculty's id: jo7894

John Miller jo7894

Engineering Department, Professor

1. Enter the information of a faculty
2. Enter information of a students
3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **6**

Enter the Staff's id: **ha5879**

No Staff member matched!

1. Enter the information of a faculty
2. Enter information of a students

3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **5**

Name of the staff member: **Jamal Kareem**

Enter the id: **ja6980**

Department: **English**

Status, Enter P for Part Time, or Enter F for Full Time: **f**

Staff member added!

1. Enter the information a faculty
2. Enter information of a students
3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **6**

Enter the Staff's id: **ja6980**

Jamal Kareem ja6980

English Department, Full Time

1. Enter the information a faculty
2. Enter information of a students
3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **7**

Enter the id of the person to delete: **ab1234** **(If ID exists then remove that person from your array)**

Sorry no such person exists.

1. Enter the information a faculty
2. Enter information of a students
3. Print tuition invoice
4. Print faculty information
5. Enter the information of a staff member
6. Print the information of a staff member
7. Delete a person
8. Exit Program

Enter your selection: **8**

Would you like to create the report? (Y/N): **y**

Would like to sort your students by descending gpa or name (1 for gpa, 2 for name): **1**

Report created and saved on your hard drive!

Goodbye!

Sample Report (**report.txt**)

Report created on 07/08/2023

(Your code should print the date of the day we run it)

Faculty Members

1. John Miller
ID: jo7894
Professor,Engineering

Staff Members

1. Jamal Kareem
ID: ja6980

English, Full Time

Students (**Sorted by gpa in descending order**)

1. Julia Alvarez

ID: ju1254

Gpa: 3.26

Credit hours: 7

2. Matt Jones

ID: ma0258

Gpa: 2.78

Credit hours: 0