Computing and Algorithms I

Project 6

CS–101

Fall 2017

In this project, you will write a program to prepare reports for a small database on people. This lab is worth 50 points for design, 50 points for the code.

Class Hierarchy

You will use a class hierarchy to practice inheritance and polymorphism. You will define a class called `Person` which may or may not be `abstract`. `Person` will have two subclasses, `Student` and `Employee`. There will be two subclasses of `Student`, namely `GraduateStudent` and `UndergraduateStudent`. Likewise, `Employee` will have two subclasses, `Faculty` and `Staff`. In addition, you will create a Date class which you will use to store dates. The Date class will have a compareTo method similar to the one for the String class, which you will use to order dates. The dates will be ordered by year, then by month for dates with the same year, then by day of the month for dates with the same year and month.

The Database

The database consists of records for four kinds of people: GraduateStudent, UndergraduateStudent, Faculty, and Staff. UnderGraduateStudents have the following attributes (`string` is a String which does not contain the character #, `d` is a digit, `a` is an alphabetic character, `N` is an alphanumeric character (either a digit or an alphabetic character), all others, such as dashes "-", are literal):

- name: `string`
- address: `string`
- phone number: `(ddd)ddd-dddd`
- e-mail address: `NNNNNNNN@aaaaaaaa.aaa`
- birth date: `dd/dd/dddd`
- status: `a`

  Note that status will have an input value of f for freshman, s for sophomore, j for junior, or r for senior.

GraduateStudents have the following attributes.

- name: `string`
- address: `string`
- birth date: `dd/dd/dddd`
- phone number: `(ddd)ddd-dddd`

- e-mail address: `NNNNNNNN@aaaaaaaa.aaa`
- status: `a`
- assistantship type: `a`

  Note that status will have an input value of m for master, or d for doctoral. The assistantship type will have an input value of t for teaching or r for research.

Faculty have the following attributes.

- name: `string`
- address: `string`
- phone number: `(ddd)ddd-dddd`
- e-mail address: `NNNNNNNN@aaaaaaaa.aaa`
- office: `string`
- salary: `dddddd.dd`
- hiring date: `dd/dd/dddd`
- title: `string`
- office hours: `string`

Staff have the following attributes.

- name: `string`
- address: `string`
- title: `string`
- phone number: `(ddd)ddd-dddd`
- e-mail address: `NNNNNNNN@aaaaaaaa.aaa`
- office: `string`
- salary: `dddddd.dd`
- hiring date: `dd/dd/dddd`
- supervisor name: `string`

  Note that all the input dates are in the order month/day/year.

The Input

Input will come from a **file**, the name of which is specified as a **command line argument**. The data will be stored one person per line, with a single # character separating each field. The very first character of each line will be either "u" (for UndergraduateStudent), "g" (for GraduateStudent), "f" (for Faculty) or "s" (for Staff), followed by a #, followed by the fields, *in the order given above for each type of person.* Note that no field may have a # in it. The beginning character of the input line will not be stored in any instance variable of any of the objects defined in the hierarchy.

The Output

Your output will consist of:

1. Title
2. Identification information (Your name, course and section number),
3. A heading stating that the next group of output lines is an echo print of the input file. After this heading, your program will print an echo of the input (each line of the file will be printed to output as is),
4. The entire database, *sorted by name*, with its own header,
5. The staff, *sorted by date*, with its own header,
6. The employees, *sorted by salary*, with its own header, and
7. The graduate students *sorted by address*, with its own header.

   After the echo print, you are printing values of each object. Each object has instance variables as specified above. These records need to be printed in an easy to read format. For each object, you will print the type of object (such as Faculty) on its own line. Each field will be printed with a label on its own line, with the line tabbed in from the type on the first line. Your program will print a blank line between each object output. For example:

```
Faculty
    name: Livingston Seagull, Ph.D.
    address: 123 Avian Avenu, Miami FL 56732
    phone number: (505)864-3891
    e-mail address: seag7245@univmiam.edu
    office: 543 Main Building
    salary: $65,812.01
    hiring date: July 1, 1993
    title: Associate Professor of Flight Engineering
    office hours: Saturdays midnight til 1 a.m.
```

   Note that the date is output in the above format, month name is spelled out, followed by a space, the day, a comma, a space, then the year.

Requirements

You will create one and only one array of Person, no arrays of Employee, Staff, etc. Your program will run for any valid input of 1 to 100 people. This lab will demonstrate that you understand polymorphism using a class hierarchy.

There will be exactly one constructor for each of the classes UndergraduateStudent, GraduateStudent, Faculty, and Staff. The parameters for these constructors will be the values constructed from the input fields. Note that the parameter for a date input will be a Date object which you will construct from the date input field.

Deliverables

This assignment will be your final project. The design and code are due by the beginning of class on Monday of week 11. All of your java code will be zipped into a single file named p6.zip and submitted to blackboard before the beginning of that class period.

The design will include UML class diagrams, one for each class you have created for your solution. You will also write a data table for the instance variables for each class. In addition, for each method, there will be an algorithm and a data table. Algorithms for constructors, accessor methods, and mutator methods will not be required for this project. Furthermore, there will be a UML interaction diagram for this project showing hierarchy and any invocation or instantiation interactions. Note that the design is a separate document from the code, and you will most likely use a spreadsheet program in creating this document.

Prepare the following items to turn in via blackboard:

1. Your design for this lab which includes all UML class diagrams, legend, and UML class interaction diagram. This also includes all required algorithms and data tables. Upload your design to blackboard area Project6a.

2. Zip together the following files and upload them as file p6.zip to blackboard area Project6b:

   - A listing of your source code. Note that the algorithms and data tables from the design will be included as comments in the source code.
   - Your input file with at least 20 correct lines of people input, a minimum of 4 of each type.
   - A file which is the output obtained by running your code on the your input file.

Notes

1. You must write your own sorting routines for this program (using any material from the text or from class notes.) You may use any sorting algorithm you know, provided you write the code yourself.

2. You may assume there will be no more than 100 people in the database.

3. The typical ways to break an input line into its fields is to use a Scanner object on the input line String, changing the delimiter appropriately, or to use the split method of a String object.