

CPSC 231: Computer Science II
Fall 2020

Programming Mastery Project 4a: Inheritance, Interfaces, & Abstract Classes – Oh My!
Suggested Completion Date: Wednesday, December 2, 2020 at 11:59pm

Overview

The university has approached you about building a database for the university. The database consists of three types of Chapman Affiliates (represented as Abstract classes) – Faculty, Students, and Staff. They have the following attributes:

Affiliates: name, age, address, phone number, year came to Chapman

Faculty: faculty id, department, yearly salary, number of papers

Students: student id, major, minor, class standing

Staff: staff id, title, building

There are 3 types of faculty members: Assistant Professors, Associate Professors, and Full Professors. They have the following attributes:

Assistant: years until tenure

Associate: years since tenure

Full: years until retirement

There are 2 types of students: Graduate and Undergraduate

Graduate: numbers of papers published, thesis advisor

Undergrad: number of courses taken, scholarship amount

There are 2 types of staff: Full time and Part time

Full time: yearly salary

Part time: hourly salary

Your classes should take advantage of inheritance to minimize duplication of code. All classes should have necessary constructors, mutators, and accessors. All classes should be documented thoroughly using Javadoc. The goal of the assignment is to make these classes as easy to reuse as possible.

In addition to the above, all Affiliates implement 2 interfaces: Comparable and Printable.

Comparable is a Java interface. You should implement the compareTo method to use the year the affiliate came to Chapman to sort. The longer they have been at Chapman, the higher they are on the seniority list. **YOU SHOULD NOT BE CREATING YOUR OWN COMPARABLE INTERFACE.**

Printable is an interface you design. It should contain a single method, `print()`, that prints ALL information associated with the class instance.

Implementation details to follow in MP4b.

Deliverables

None. Your source code will be delivered as part of MP4b.

CPSC 231: Computer Science II
Fall 2020
Programming Mastery Project 4b: Files, Exceptions, & HashMaps – Oh my!
Due: Thursday, December 10, 2020 at 11:59pm

Overview

Having set up your beautiful class structure design, let's put your database into action. Your job is to implement the following in an AffiliatesDriver class.

When your program is run it should display the following options:

- 1) Create an affiliate record
- 2) Print information for an Affiliate given the id
- 3) List all affiliates in order of seniority
- 4) Delete a record given the id
- 5) Save your database to a file
- 6) Restore your database from a file
- 7) Exit

Each of the options should prompt for the required information as needed.

To save your database to a file, the program should prompt for the desired output file name. The program should then write the information for each affiliate record currently in your database to the file in plain text, ONE AFFILIATE PER LINE, with each attribute value separated by commas. Start each line with the type of affiliate (the class name), followed by a comma. The attributes should then follow in alphabetical order (within class attributes, not across classes), separated by commas, starting at the top of the class hierarchy and moving down.

To restore your database from a file, the program should prompt for the desired input file. The format of the input file will be the same as described above for the output file. Once the file is provided, the program will create affiliate records for each entry (line) in the input file and add them to the database.

Hint: Use inheritance and polymorphism to simplify saving to and restoring from files.

Your database should be implemented as a HashMap that maps id numbers (integers) to instances of Affiliate objects.

Your code should handle all exceptions (e.g. for file processing) appropriately.

Deliverables

Submit your Javadoc commented source files (for part a and b) and a README in a zip file by the due date.