Joe Del Rocco
jdelrocco [at] stetson [dot] edu
Assistant Professor of Practice, Computer Science
Stetson University
421 N Woodland Blvd, DeLand, FL, 32723
www.stetson.edu

CSCI 142
(all sections)
Spring 2022
Assignment

Assignment 1

Due: Monday 2/7/2022 11:59pm

Contents

Program

This program is meant to give you practice with Generics, Iterators and the List ADT. You will load 2 separate provided files. The first one contains numbers which you will load into an `ArrayList` of type `Double`. The second one is a comma-separated-values list of students, which you load into a `LinkedList` of type `Student`, a custom class that you will create. You will display the list of numbers and the list of students using the same method implemented with Generics. You will also use OOP Abstraction by having the `Student` class implement the `Comparable` interface. By doing so, you will be able to use the `Collections.sort()` method on your list of `Student` objects to effectively sort them. Finally, you will code a method to remove students that have grades less than an A (90%). See the Example Output for an example of how this program should perform.

Here is the link to the GitHub Classroom assignment:
https://classroom.github.com/a/_mlKYmZw

1) TODO

There are various `// TODO:` comments provided in your assignment source code. You don't have to follow them exactly, but they are how we implemented our solution. Note there is a TODO window in IntelliJ IDEA which can be used to quickly jump between them as well.

2) Student

Create a `Student` class with 3 encapsulated member variables for: student last name, first name, and grade. The constructor should take 3 arguments: a last name, a first name, and a grade, and then assign those to the private member variables. Add accessor (getter) methods for each of the 3 member variables. Override the `toString()` method that comes from the `Object` class by returning a full string describing the student (e.g. "[first name] [last name] - [grade]").

3) Load Students

Uncomment the method provided for loading the .csv roster file of students provided. Call this method from `main()` to load all of the students and return a `LinkedList` of type `Student`. Create an equivalent variable in `main()` to hold the list of students that come back from the method.

4) Load Constants

At this point you should do the same for the file of constants (numbers). Create a new method to load the constant data file provided and return an `ArrayList` of type `Double` which contains all the constants from the provided text file. You should know be able to figure out how to code this method on your own, which is simpler than the one for loading students, but if not, watch this video on how to load a simple text file and read line by line. You will have to convert each line that you read from the file from type `String` to type `Double`, then `add()` each one to the `ArrayList`. Call this method from `main()` to load the list of constants into an equivalent variable in `main()`.

Tip

Recall that you can convert `String`s to `Double`s like this:

```
Double d = Double.valueOf([any string]);
```

5) printStuff()

Next, write a method called `printStuff()` that uses Generics and takes a `List` of any type. This means you can pass both your `ArrayList` of `Double`s and your `LinkedList` of `Student`s to the same function.

It will print the `List` out with each element on its own line. Call this method from `main()` to print out both your lists.

6) Comparable

Next, implement the `Comparable` interface from the `Student` class. The interface only has one single method, `compareTo()`. Override it in the `Student` class. If you need help with overriding a method, you can consult your book (Section 8.2 pp. 373-376 of the 5th edition). You can choose how to sort `Students` however you wish: by first name, last name, or grade. By implementing `Comparable`, you can now call `Collections.sort()` from `main()` and pass it your list of students.

7) trimStudents()

Finally, write a method called `trimStudents()` which takes the `LinkedList` of students and a grade to filter by, and uses an `Iterator` of type `Student` to iterate through the passed in list and remove any `Student` with a grade less than the filter passed in. If you need another explanation of iterators, watch this video. Call this method from `main()` passing the list of students and a grade filter of 90.0. After you have trimmed the student list, call `printStuff` again and pass the list of students.

Submission

You will commit and push your changes to your specific GitHub Classroom repository for this assignment. You are encouraged to use an IDE for development, but we will compile and run your program using the shell/terminal during grading, so it isn't a bad idea to test it in that environment to make sure it works. Please follow the directions in this assignment, make the requested code changes, and commit and push your changes any time before the due date. Please see the advice below; it is important for grading purposes. **Failure to follow these directions will result in a loss of points.**

Always make sure to:

- Keep all source files in the folder called `src`, which is one directory in from the root of your repo
- Do not commit multiple copies of the same named source file; modify the ones provided to you. In other words, do not make an old and new version of the same file
- The main starting source file should always be called `Main`
- When loading resources, do not use absolute paths to files on your drive; use relative paths
- Do not have the keyword `package` at the top of any files. Some IDEs add your files to a custom package by default. Please remove this line, as it complicates grading.

Rubric

Task	Percentage
Assignment files not pushed to GitHub	Grade is 0%
General attempt at solving the assignment	40%
Student class: constructor, encapsulated	20%
Loading and displaying constants properly	5%
Loading and displaying students properly	5%
Implementing Comparable interface and sorting students	10%
Using Generics to print both lists	10%
Using Iterator to remove elements from student list	10%
Total	100%

Example Output

```
Windows PowerShell
Constants:
3.14159
1.61803
1.38064852E-23
9.81
6.67408E-11
2.99792458E8
42.0
6.02214076E23

Students (unsorted):
Grace Hopper - 80.0
Edsger Dijkstra - 90.0
Ada Lovelace - 60.0
Bill Joy - 70.0
Jean Bartik - 70.0
Donald Knuth - 90.0
Claude Shannon - 100.0
Alan Turing - 100.0
Barabara Liskov - 80.0

Students (sorted):
Ada Lovelace - 60.0
Alan Turing - 100.0
Barabara Liskov - 80.0
Bill Joy - 70.0
Claude Shannon - 100.0
Donald Knuth - 90.0
Edsger Dijkstra - 90.0
Grace Hopper - 80.0
Jean Bartik - 70.0

Students (only A's):
Alan Turing - 100.0
Claude Shannon - 100.0
Donald Knuth - 90.0
Edsger Dijkstra - 90.0
```