San José State University
Computer Science Department
CS151, Object Oriented Design and Programming, 05, Fall 2022

Homework #4

Objective:

This homework's objective is to review and understand the unit on Java collections and how to use them in your Java application.

Details:

Exercise 1:

Define and implement a class that works with Java ArrayList<Integer> data structure. This class should have the following methods:

Method declaration	**Description**
ArrayList<Integer> sort(ArrayList<Integer> myLst, Boolean ascending)	Accepts an array list of integers and sorts it in ascending or descending order. Make sure to account for cases with duplicate values. The method returns the sorted list.
ArrayList<Integer> swapLargestSmallest(ArrayList<Integer> myLst)	Accept an array list of integers and swamps positions of the largest and the smallest elements. If multiple values representing largest/smallest elements are present then pick the first position. The method returns the modified list.
void table(ArrayList<Integer> myLst)	Accepts an array list of integers, creates a table of unique values and the number of times each value occurs, sorts this table, and outputs it to command line.

Save this class and its definition into a file named **ListManipulator.java**.

I will test your class with my own tester class using the API specification provided above.

Exercise 2:

Define and implement class **Course**. This class should contain the following fields: course name, course description, department, time the course starts, weekday the course is held on (for simplicity, let us assume the course only meets once a week). This class should contain getters and setters for all its attributes. This class also needs at least one constructor. Save this class and its definition into a file named **Course.java**.

Define and implement class **Student**. This class should contain the following fields: first name, last name, age, gpa, major, department, courses. Age should be an integer value. GPA should be a floating point value. Courses should be a linked list of Course variables. Class Student should contain getters and setters for all its attributes. This class also needs at least one constructor. In class Student implement the following methods: addCourse() to add a new course, removeCourse() to remove a course from this student, sortCourses() to print out a sorted list of courses. Method sortCourses() should accept parameters to specify whether the sorting should be ascending or descending and also based on which course attribute to sort the courses. The output should be printed to command line standard output. Save this class and its definition into a file named **Student.java**.

I will test your class with my own tester class using the API specification provided above.

Exercise 3:
Define and implement class **Person**. This class should contain the following fields: first name, last name, and age. Age should be an integer value. This class should contain getters and setters for all its attributes. This class also needs at least one constructor. Save this class and its definition into a file named **Person.java**.

Define and implement class **PersonTest**. This class should implement *main()* method. In the body of the *main()* method you should create an array list of persons with the following Person instances:
Joe Smith, 40
Amy Gold, 32
Tony Stork, 21
Sean Irish, 24
Tina Brock, 28
Lenny Hep, 18

Sort your list of persons based on their age, both ascending and descending. Sort your list of persons based on their first name, both ascending and descending. Sort your list of persons based on the length of their last name, both ascending and descending. For each output, print the list out to command line standard output. These sort operations should produce 6 different list outputs. Save this class and its definition into a file named **PersonTest.java**.

Submission:

Submit all files created by you for the homework exercises: ListManipulator.java, Course.java, Student.java, Person.java, PersonTest.java and any other files you completed for this assignment, if any.

Make sure to submit by 11:59pm on the due date listed in Canvas. Submit your solution via Canvas.

If you have any questions, message me or the grader or both:
Yulia.Newton@sjsu.edu
madhujitaranjit.ambaskar@sjsu.edu

Grading:

Your code must compile and execute successfully in order to get full credit for this assignment. For exercise 1 and 2, I will test your class with my own tester class using the API specification provided in the exercise. For exercise 3, I will compile and execute the files.

- Program with no compile errors
- Program executes
- Program outputs what is required by the exercise

A total of 30 points are possible for this homework assignment.