OOP Lab 8: More on Collections

Sharan R Y, Madhur Kalia, Akhil Tarikere

General Instructions:

1) Read the question carefully.
2) Code according to the order of test cases to minimize wasted efforts.
3) Indent your code to make it readable and easier to debug.
4) Use the right data structures at the right place to get marks.

In this lab, you will be learning about various kinds of collections by designing a library management system. Since Waves ‘19 was hosted last week’s lab and the festivities have finished, it is time for us students to get back into the academic season. The library in our very own BITS Pilani, Chandigarh campus does not have a library management system which is causing havoc for everyone involved. You have been tasked with making the library management system with the given specifics. All the students’ hopes for doing well in their exams lies in your hands assuming they actually use the library for study purpose.

In this lab, the tasks allotted to you will focus on getting your concept of collections right. In this system, we have 3 classes- Library, Shelf and Book. Every Library is made up of shelves and every shelf is made up of book(s). You will have to implement various methods in each of these classes as per the Javadoc so as to successfully inaugurate the exam season.

Test Cases:

Test	Function Name	Class	Marks
1.	Book's constructor and getters	Book	1
2.	compareTo & equals	Book	1
3.	getShelf Note: Make sure to complete Shelf class for this.	Book & Shelf	1
4.	Shelf's constructor and getters	Shelf	1
5.	addBook	Shelf	1
6.	organize	Shelf	1
7.	clearShelf	Shelf	1
8.	isRedundant	Shelf	1
9.	Library's constructor and getters & addShelf	Library	1
10.	searchBook	Library	1

Pointers to keep in mind:

1. Make sure to import java.io.* and java.util.*

2. Run it by creating a folder java and putting all three .java files in it.
   When you want to run it, type the following command in the terminal:

   java -jar test.jar

   (Remember to put test.jar in the same folder as 'java')
   Please type this out (don't copy it from the PDF).

3. You might not get marks for some of the other methods if your constructors are not correct.

4. Some of the test cases may depend on some test cases above it (numerically), so try to write your code in the numerically ascending order to match the test cases.