Data Structures and Programming
Final Exam

Final Exam Programming Question (80 points)

You are given a Java program that manages the operations of a fitness center. The center has a list of training sessions, a list of members, and a list of users. The list of users is provided in the file **users.txt**. Each user has an id, username, and password. Users with an ID that starts with the letter "M" are members of the gym who can enroll in training sessions. Users with an ID that starts with the letter "A" are administrators who can view the income of the gym and the popular training sessions. The list of members is provided in the file **members.txt**. Each member has an ID, name, and a list of training sessions he/she is enrolled in. The list of training sessions is provided in the file **sessions.txt**. Each session has a code, title, time in minutes, fee in dollars, and the number of members enrolled in the session. The program allows the users to login using a username and password. Once the login credentials are validated, the program will allow the users, who are members, to view the training sessions or add/remove training sessions to/from their schedule. The program will allow the users, who are administrators, to view the gym income from the member contributions and the total income, and the list of training sessions ranked by popularity.

The given program is at 60% of its completion. Your task consists in completing the missing parts as they are described below.
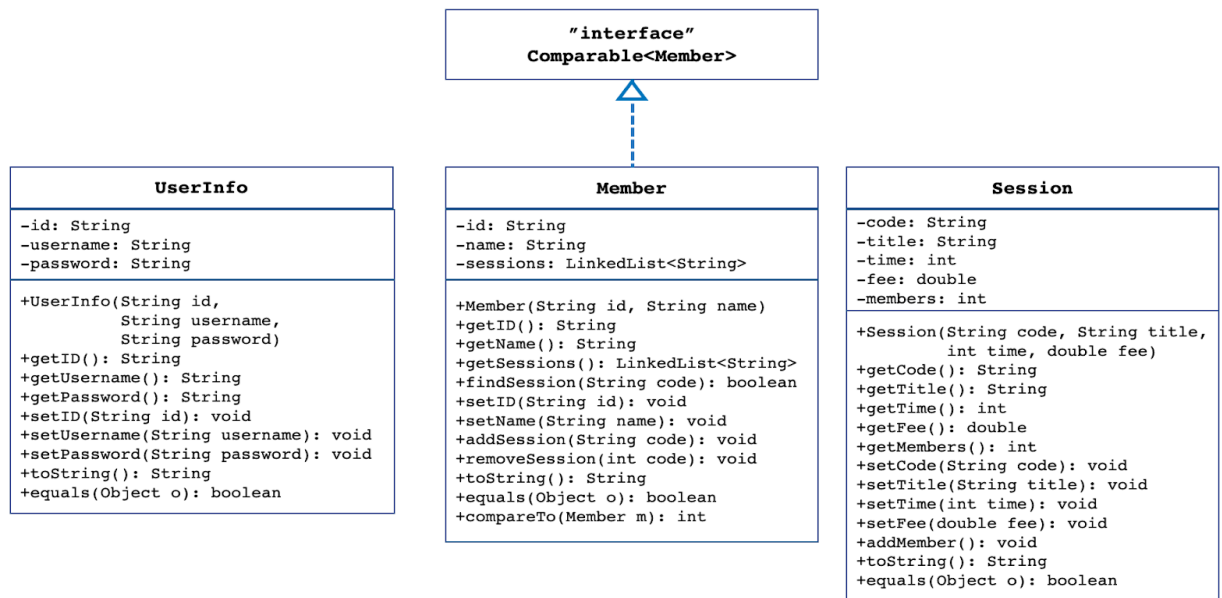
Complete the following tasks to deliver a working program for the fitness center similar to the sample run provided at the end of this document.

1. Write the definition of the following method in the class **BST**.

   a. **ArrayList<E> toList()**: This method is <u>**recursive**</u> and returns an array list with the values of the BST nodes traversed in order. **(5 points)**

   Determine the time complexity of the method **toList()** using the big-O notation. Add the time complexity as a comment before the method's header.

2. Use the definitions of the classes **UserInfo**, **Member**, and **Session** as provided and illustrated in the UML diagram below. Go over the code of each class to understand the attributes and the interface of each class.

```
                              ┌─────────────────────────────┐
                              │        "interface"          │
                              │   Comparable<Member>        │
                              └─────────────────────────────┘
                                           △
                                           ┊
```

UserInfo
-id: String
-username: String
-password: String

+UserInfo(String id,
      String username,
      String password)

+getID(): String
+getUsername(): String
+getPassword(): String
+setID(String id): void
+setUsername(String username): void
+setPassword(String password): void
+toString(): String
+equals(Object o): boolean

Member
-id: String
-name: String
-sessions: LinkedList<String>
+Member(String id, String name)
+getID(): String
+getName(): String
+getSessions(): LinkedList<String>
+findSession(String code): boolean
+setID(String id): void
+setName(String name): void
+addSession(String code): void
+removeSession(int code): void
+toString(): String
+equals(Object o): boolean
+compareTo(Member m): int

Session
-code: String
-title: String
-time: int
-fee: double
-members: int
+Session(String code, String title,   int time, double fee)
+getCode(): String
+getTitle(): String
+getTime(): int
+getFee(): double
+getMembers(): int
+setCode(String code): void
+setTitle(String title): void
+setTime(int time): void
+setFee(double fee): void
+addMember(): void
+toString(): String
+equals(Object o): boolean

3. The test class **GymManager** is provided with four completed methods (**main()**, **readSessions()**, **adminOperations()**, and **login()**) and five empty definitions for the methods you will write. Go over the code of the completed methods to understand the overall behavior of the program and complete the following:

    a. Write the definition of the method **readUsers()** to read the file **users.txt** into the hash table **users**. Note that the field **username** is used as the key in the hash table and the object of type **UserInfo** as the value. **(10 points)**

    b. Write the definition of the method **readMembers()** to read the file **members.txt** into the binary search tree **members** and update the list **sessions** (call **addMember()** on each training session a member is enrolled in). **(10 points)**

c.   The definition of the method **login()** is given. The method throws an exception of type **InputMismatchException** when the username is not valid. Add a **try-catch** block, in the main method, to handle the exception thrown by the method **login()**. **(5 points)**

d. Complete the definition of the method **memberOperations()** for the gym members to view, add, or remove training sessions. See the sample program run for the behavior of these operations. **(20 points)**

e. The method **adminOperations()** for the gym administrators displays the weekly income and the training sessions ranked by popularity. This method calls the two methods described below.

1. Write the definition of the method **printIncome()** to print the income of the gym by showing the total payment for each member and the total weekly income for the gym. **(20 points)**

2. Write the definition of the method **printSessions()** to print the training sessions sorted in descending order of the number of members enrolled in the session. Use the given **mergeSort** method to sort the array list **sessions**. To perform this task, you may decide to modify either the class **Session** or the method **mergeSort**. **(10 points)**

Submit the following files on course site:

HashMap.java,

BST.java,

UserInfo.java,

Member.java,

Session.java,

MergeSort.java, and

GymManager.java

3

 Sample runs of the program are provided below to test your program.

Sample RUN 1: Invalid username
```
---------------------------------------------------------------------
Enter username: lfe
Invalid username (must be 3 letters followed by 3 digits).
```

Sample RUN 2: Incorrect username or password for three times
```
---------------------------------------------------------------------
Enter username: lfe212
Username not found. Try again.

Enter username: lfe218
Enter password: }BM+]D_R4$qO1
Password incorrect. Try again.

Enter username: lfe218
Enter password: }BM+]_D_R4$qO1
Password incorrect. Try again.

Incorrect username or password more than 3 times.
Terminating the program...
```

Sample RUN 3: The logged in user is a gym member
```
---------------------------------------------------------------------

Enter username: lfe218
Enter password: }BM+]`D_R4$qO1

Select an operation:
1: View My Training Sessions
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
1
Code       Title                     Duration (minutes)  Cost
CF2        Cardiac_Fitness_Level_2   90                  $250.00
C          Cycling                   60                  $110.00

Select an operation:
```

```
1: View My Training Sessions
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
2
```

Code	Title	Duration (minutes)	Cost
B&B	Bike&Barre	45	$100.00
BC	Boot_Campo	30	$180.00
CF1	Cardiac_Fitness_Level_1	45	$130.00
CF2	Cardiac_Fitness_Level_2	90	$250.00
CS	Core_Synergy	60	$100.00
C	Cycling	60	$110.00
GY	General_Yoga	30	$85.00
AY	Advanced_Yoga	90	$200.00
T	Tabata	45	$100.00
Z	Zumba	60	$120.00

```
Select an operation:
1: View My Training Sessions
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
3
```

Code	Title	Duration (minutes)	Cost
B&B	Bike&Barre	45	$100.00
BC	Boot_Campo	30	$180.00
CF1	Cardiac_Fitness_Level_1	45	$130.00
CF2	Cardiac_Fitness_Level_2	90	$250.00
CS	Core_Synergy	60	$100.00
C	Cycling	60	$110.00
GY	General_Yoga	30	$85.00
AY	Advanced_Yoga	90	$200.00
T	Tabata	45	$100.00
Z	Zumba	60	$120.00

```
Enter the code of the session you want to add
Z
Training session added successfully.
```

```
Select an operation:
1: View My Training Sessions
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
1
```

Code	Title	Duration (minutes)	Cost
CF2	Cardiac_Fitness_Level_2	90	$250.00
C	Cycling	60	$110.00
Z	Zumba	60	$120.00

```
Select an operation:
1: View My Training Sessions
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
4
```

Code	Title	Duration (minutes)	Cost
CF2	Cardiac_Fitness_Level_2	90	$250.00
C	Cycling	60	$110.00
Z	Zumba	60	$120.00

```
Enter the code of the session you want to remove
C
Training session removed successfully.

Select an operation:
1: View My Training Sessions
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
1
```

Code	Title	Duration (minutes)	Cost
CF2	Cardiac_Fitness_Level_2	90	$250.00
Z	Zumba	60	$120.00

```
Select an operation:
1: View My Training Sessions
```

```
2: View All Training Sessions
3: Add Training Session
4: Drop Training Session
5: Logout
5
Thank you for your visit!
```

Sample RUN 4: The logged in user is an administrator

```
Enter username: haa212
Enter password: Z7Ty;g#;iYC7|
```

Member ID	Member name	Sessions	Fees
M100029	Eun,Coody	[CF1, BC]	$310.00
M100181	Willow,Kusko	[BC, Z, GY]	$385.00
M100204	Dominique,Dickerson	[GY, B&B, BC]	$365.00
M100279	Lucy,Treston	[T, Z, BC]	$400.00
M100280	Salome,Lacovara	[B&B, GY, AY]	$385.00
M100283	Elke,Sengbusch	[BC, T, CS]	$380.00
M100556	Janine,Rhoden	[T, GY, B&B]	$285.00
M101050	Diane,Devreese	[CF1, CS]	$230.00
M101071	Yoko,Fishburne	[Z, BC]	$300.00

```
 .
 .
 .
```

M197804	Mona,Delasancha	[AY, AY]	$400.00
M198041	Teddy,Pedrozo	[B&B, AY, GY]	$385.00
M198069	Donte,Kines	[Z, CF2]	$370.00
M198174	Delmy,Ahle	[C, GY]	$195.00
M198209	Catarina,Gleich	[CF2]	$250.00
M198213	Lynelle,Auber	[CF1, B&B, CS]	$330.00
M199209	Gracia,Melnyk	[BC, CS, CS]	$380.00
M199317	Yolando,Luczki	[BC]	$180.00
M199734	Rima,Bevelacqua	[CF1, CF2, GY]	$465.00
M199932	Izetta,Funnell	[Z, Z, CS]	$340.00
Total Income			$133090.00

```
Session Title                     # Enrolled Members
```

```
Zumba                     113
General_Yoga              106
Bike&Barre                98
Cardiac_Fitness_Level_2   98
Boot_Campo                96
Core_Synergy              96
Tabata                    96
Cycling                   95
Cardiac_Fitness_Level_1   93
Advanced_Yoga             86
```

```
Zumba                     113
General_Yoga              106
```