

Due Date

Monday, October 3, 11:59pm.

Submission

1. Submit a **single zipped file** to Canvas. The zipped file **MUST** include the following grading items.
 - **Source folder src**, containing the source code *.java files [60 points]
 - (a) Create a package to organize the source files; use all lowercase letters for the package name; you will **lose 2 points** if this is not done properly.
 - (b) **MUST** include all team members' names using the **@author** tag in the comment block on top of EVERY Java class, or you will **lose 2 points**.
 - **Test design document**. You must use a document editor and write all test cases you use to test the **isValid()** method of the Date class, and the **compareTo()** method of the Member class. [15 points]
 - **Testbed main()** in the following classes implementing the test cases in the test design document above.
 - (a) Date class [10 points]
 - (b) Member class [10 points]
 - **Javadoc** folder in which you must include all the files generated by the Javadoc. [5 points]
2. The submission button on Canvas will disappear after **October 3, 11:59pm**. It is your responsibility to ensure your Internet connection is good for the submission. **You get 0 points** if you do not have a submission on Canvas. **Projects sent through the emails will not be accepted.**

Project Description

A fitness chain has gyms at 5 different locations in the central New Jersey area:

Bridgewater, 08807, Somerset County
Edison, 08837, Middlesex County
Franklin, 08873, Somerset County
Piscataway, 08854, Middlesex County
Somerville, 08876, Somerset County

Your team will develop a simple software to help the fitness chain manages the gym memberships and fitness class schedules. The initial phase is to provide a console-based interactive user interface to process the operations in command lines and display the results. A command line begins with the operation the user needed to perform followed by the data tokens to complete the operation. An operation will be represented with a single character or two-character uppercase letters. The operation code and the data tokens will be delimited by spaces in a command line. The operation code is **case-sensitive**, which means operation codes with lowercase letters should be rejected by the software.

The initial requirement for the software is to provide the following functionalities:

- (1) Add member information to the member database. Each member is uniquely identified by their first name, last name, and date of birth. A membership has an expiration date and a specified gym location. A member must be 18 or older to join.
- (2) Remove member information from the member database; for simplicity, members who want to cancel the membership will be removed from the database.
- (3) Display the list of members sorted by last and first name OR sorted by county and zip code OR sorted by the expiration date, in ascending order.

- (4) Members must check in for the online fitness classes. Regardless of the membership locations, as long as their memberships have not expired, they can check in any online fitness classes happening the day of checking in. Some of the online fitness classes may be holding at the same time, the software should check if there is a time conflict when the members are checking in. Currently, the fitness chain offers the following online fitness classes on a daily basis.

Fitness Class Name	Instructor	Time
Pilates	Jennifer	9:30
Spinning	Denise	14:00
Cardio	Kim	14:00

To provide the above functionalities, the operation codes at the beginning of the command lines are listed below.

- **A** command, to **add a member** to the member database. Data needed to add a member includes member's first and last name, date of birth, membership expiration date and location (in city name) of the gym membership. Below is a sample command line for add a member. You can assume that the user will always enter enough data tokens to add a member in the format shown below.

A April March 3/31/1990 6/30/2023 Piscataway

The above command line starts with the A command, followed by the member's first name, last name, date of birth, membership expiration date, and the city name identifying the gym location. The dates shall be given in **mm/dd/yyyy** format. The names and locations are NOT case-sensitive, that is, April March and april march are the same person as long as the dates of birth are also the same; PISCATAWAY and Piscataway represent the same location. The user can enter an expired membership date. Human errors are very common. Your software shall provide certain degree of reliability and shall not allow the member be added to the database if the data tokens in the command line have the following conditions.

- Any date that is not a valid calendar date
 - The date of birth is today or a future date
 - A member who is less than 18 years old
 - An invalid city name, that is, the gym location doesn't exist
- **R** command, to **cancel the membership** and remove the specified member from the database, for example,
R Paul Siegel 5/1/1999

The above command line removes the member from the database. Your software must reject if the user is removing a non-existing member.

- **P** command, to display the list of members in the database without sorting (current order in the array.)
- **PC** command, to display the list of members in the database ordered by the county names and then the zip codes; that is, if the locations are in the same county, ordered by the zip codes.
- **PN** command, display the list of members in the database ordered by the members' last names and then first names; that is, if two members have the same last name, ordered by the first name.
- **PD** command, display the list of members in the database ordered by the expiration dates. If two expiration dates are the same, their order doesn't matter.
- **S** command, display the fitness class schedule. A fitness class shall include the fitness class name, instructor's name, the time of the class, and the list of members who have already checked in today. For simplicity, assuming the schedule is for "today" only, you do not need to handle a multiple-day schedule.

- **C** command, for members check-in, for example, the command line to check in Pilates.

`C Pilates Mary Lindsey 12/1/1989`

Your software shall not allow a member to check in if

- the membership has expired
 - the member does not exist
 - the date of birth is invalid
 - the fitness class does not exist
 - there is a time conflict with other fitness classes
 - the member has already checked in
- **D** command, to drop the fitness classes after the member checked in to a class, for example,

`D Pilates Mary Lindsey 12/1/1989`

Your software shall not allow the member to drop the class if the member is not checked in, or the date of birth is invalid, or the fitness class does not exist.

- **Q** command, to stop the program execution and display "Gym Manager terminated.", so the user know that your software has stopped running.

Project Requirement

1. You **MUST** follow the Coding Standard and Ground Rules posted on Canvas under Week #1 in the “Modules”. **You will lose points** if you are not following the rules.
2. You are responsible for following the Academic Integrity Policy. See the **Additional Note #13** in the syllabus.
3. There are test cases in the file **Project1_testCases.txt** for you to test your project. The associated output is listed at the end of this document for your reference. The graders will be using the sample test cases to test your project. Your project should be able to take the test cases from the console in the same order without getting any exceptions and without terminating abnormally, including the empty lines. You will **lose 2 points** for each incorrect output or each exception causing the project to terminate abnormally. You should create an instance of the Scanner class to read from standard input.
4. Each Java class must go in a separate file. **-2 points** if you put more than one Java class into a file.
5. Your program **MUST** handle bad commands; **-2 points** for each bad command not handled, with a **maximum of losing 6 points**.
6. You are not allowed to use any Java library classes, EXCEPT the **Scanner, StringTokenizer, Calendar** and **DecimalFormat** class. **You will lose 5 points** for each additional Java library class imported, with a **maximum of losing 10 points**.
7. You are not allowed to use the Java library class **ArrayList** anywhere in the project, or use any classes in the Java Collections, or **you will get 0 points for this project**.
8. When you import Java library classes, be specific and **DO NOT** import unnecessary classes or import the whole package. For example, **import** java.util.*; this will import all classes in the java.util package. You will **lose 2 points** for using the asterisk “*” to include all the Java classes in the java.util package, or other java packages, with a **maximum of losing 4 points**.
9. You **MUST** include the Java classes below. **-5 points** for each class missing or NOT used. You **CAN** add necessary constructors, private methods (helper methods), and other public methods to each class. You should define necessary constant identifiers and **do not use MAGIC NUMBERS**. A good approach is to use an enum class or a public class to define all the constant names and their associated values.

(a) Member class

```

public class Member implements Comparable<Member>{
    private String fname;
    private String lname;
    private Date dob;
    private Date expire;
    private Location location;
    ...
    @Override
    public String toString() { }
    @Override
    public boolean equals(Object obj) { }
    @Override
    public int compareTo(Member member) { }
    ...
}

```

- You CANNOT change or add instance variables for this class. **-2 points** for each violation.
- You CANNOT read from console or use **System.out** statements in this class. **-2 points** for each violation.
- **toString()** method returns a textual representation of a member in the following format.

April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX

- **equals()** method returns true if the two first names, last names and dates of birth are equal.
- **compareTo()** method is used when sorting by names. You MUST design the test cases to thoroughly test the **compareTo()** method. As a minimum, you must include the test cases in **Project1_testCases.txt**. You must write a testbed main and implement the test cases. You must follow the instructions in the “Test Design” section in the “Coding Standard and Ground Rules”. In the testbed main, you MUST write code to print out the test results on the console showing the test cases passed or failed. The testbed main for this class **is worth 10 points**. Please use “//” comments to identify the test case numbers in the testbed main. There is no restriction on the number of lines you can have in the testbed main().
- You CANNOT change the signatures of the **toString()**, **equals()** and **compareTo()** methods. You cannot remove the **@Override** tags. **-2 points** for each violation.

(b) MemberDatabase class

```

public class MemberDatabase {
    private Member [] mlist;
    private int size;

    private int find(Member member) { }
    private void grow() { }
    public boolean add(Member member) { }
    public boolean remove(Member member) { }
    public void print () { } //print the array contents as is
    public void printByCounty() { } //sort by county and then zipcode
    public void printByExpirationDate() { } //sort by the expiration date
    public void printByName() { } //sort by last name and then first name
}

```

- This is an array-based linear data structure to hold the list of members. A new member is always added to the end of the array. An instance of this class is a growable container with an initial capacity of 4, and

automatically grows (increases) the capacity by 4 whenever it is full. The container does not decrease in capacity.

- This class shall provide the methods for managing the list of members. You CANNOT change or add instance variables. **-2 points** for each violation.
- You must implement the methods listed above, and you CANNOT change the signatures of the methods. **-2 points** for each method not implemented, signature changed or not used.
- You CANNOT use **System.out** in this class, EXCEPT the four **print()** methods, **-2 points** for each violation.
- The **find()** method searches a member in the list and returns the index if it is found, it **returns -1** if the member is not in the list. You must define a constant identifier "NOT_FOUND" for the value -1.
- The **remove()** method remove a member from the list. This method maintains the relative order of the members in the list after the remove, **-3 points** if this is not done correctly.
- You must use an **in-place sorting algorithm** to sort the list of members in the array. That is, no additional array can be declared for sorting, or you will **lose 10 points**. You must write code to implement the algorithm yourself. You CANNOT use **Arrays.sort()** or **System.arraycopy()** or any other Java library classes for sorting. You will **lose 10 points** if you do. The sorting algorithm you use can be "stable" or "not stable".
- You can add constant names and additional methods if necessary. However, the methods you added must either take no parameter or has a single parameter, which takes only an instance of Member class, such as (Member member), or you will **lose 2 points** for each violation, see the add and remove methods as an example.

(c) **GymManager class**

- This class is the **User Interface class** to process the command lines entered o the IDE console and display the results on the console. An instance of this class can process a single command line, and a sequence of command lines in batch. If it cannot process the command lines in batch, **you will lose 10 points**. This is an interactive program such that, it displays the results on the console whenever one or more command lines are entered, after the user hits the enter key.
- When your project starts running, it shall display "Gym Manager running...". In this case, the user would know the software is ready to read the command lines. It should continuously process the command lines until the "Q" command is read. That is, the "Q" command is the only way to terminate the software normally. Before the software stops running, display "Gym Manager terminated."
- You must define a **public void run()** method that includes a while loop to continuously read the command lines. You will **lose 5 points** if the run() method is missing. You **MUST** keep this method **under 40 lines** for readability, or you will **lose 3 points**. You should define necessary instance variables and private methods (helper methods) to handle different commands.

(d) **RunProject1 class** is a driver class to run your Project 1. The main method will call the **run()** method in the GymManager class.

```
public class RunProject1 {
    public static void main(String[] args) {
        new GymManager().run();
    }
}
```

(e) **Date class**

```
public class Date implements Comparable<Date> {
    private int year;
    private int month;
    private int day;

    public Date() {} //create an object with today's date (see Calendar class)
    public Date(String date) {} //take "mm/dd/yyyy" and create a Date object

    @Override
    public int compareTo(Date date) { }
    public boolean isValid() { } //check if a date is a valid calendar date
}
```

- You must implement the constructors and methods listed above. You must implement the Comparable Interface and implement the **compareTo()** method, or **lose 2 points** for each violation. You will need this method to sort by the dates.
- You CANNOT change or add instance variables, and you CANNOT use **System.out** statements in this class, EXCEPT the testbed main(), **-2 points** for each violation.
- The **isValid()** method checks if a date is a valid calendar date.
 - For the month, January, March, May, July, August, October and December, each has 31 days; April, June, September and November, each has 30 days; February has 28 days in a non-leap year, and 29 days in a leap year. DO NOT use **magic numbers** for the months, days and years. Below are some examples for defining the constant identifiers for the constants.

```
public static final int QUADRENNIAL = 4;
public static final int CENTENNIAL = 100;
public static final int QUATERCENTENNIAL = 400;
```

To determine whether a year is a leap year, follow these steps:

- Step 1. If the year is evenly divisible by 4, go to step 2. Otherwise, go to step 5.
- Step 2. If the year is evenly divisible by 100, go to step 3. Otherwise, go to step 4.
- Step 3. If the year is evenly divisible by 400, go to step 4. Otherwise, go to step 5.
- Step 4. The year is a leap year.
- Step 5. The year is not a leap year.

- You MUST design the test cases to thoroughly test the **isValid()** method. As a minimum, you must include the test cases for the dates in Project1_testCases.txt. You must write a testbed main and implement the test cases. You must follow the instructions in the “Test Design” section in the “Coding Standard and Ground Rules. In the testbed main, you MUST write code to print out the test results to the console showing the test cases are passed or failed. The testbed main for this class **is worth 10 points**. Please use “//” comments to identify the test case numbers in the testbed main. As an exception, there is no restriction on the number of lines you can have in the testbed main().

(f) **The enum classes.** You must define the following enum classes, or you will **lose 5 points** for each violation.

```
public enum Time { } //define the time of a fitness class in hh:mm
public enum Location { } //define the 5 gym locations; refer to Lecture Note #2
```

(g) **FitnessClass class**

You must include this Java class, which define a fitness class the members can check in. You can define the instance variables and methods needed. You must use the enum class Time in this class or **lose 2 points**.

10. You are required to **generate the Javadoc** after you properly commented your code. Your Javadoc must include the documentations for the constructors, private methods, and public methods of all Java classes. Generate the Javadoc in a single folder “**doc**” and include it in the zip file to be submitted to Canvas. **Please double check your Javadoc after you generated it** and make sure the descriptions are NOT EMPTY. You will lose points if any description in the Javadoc is empty. You will **lose 5 points** for not including the Javadoc.

Sample Input

Download **Project1_testCases.txt** from Canvas. These are the test cases used to generate the expected output. The graders will be using the same sequence of command lines in this file to run and grade your project.

Expected Output

Gym Manager running...

a is an invalid command!
p is an invalid command!
pc is an invalid command!
pn is an invalid command!
pd is an invalid command!
Member database is empty!
Member database is empty!
Member database is empty!
Member database is empty!
AA is an invalid command!
r is an invalid command!
RR is an invalid command!
s is an invalid command!
c is an invalid command!
d is an invalid command!
SS is an invalid command!
CC is an invalid command!
DD is an invalid command!

DOB 9/2/2022: must be 18 or older to join!
DOB 12/2/2022: cannot be today or a future date!
John Doe added.
DOB 12/20/2004: must be 18 or older to join!
DOB 2/29/2003: invalid calendar date!
DOB 4/31/2003: invalid calendar date!
DOB 13/31/2003: invalid calendar date!
DOB 3/32/2003: invalid calendar date!
DOB -1/31/2003: invalid calendar date!
Expiration date 4/31/2022: invalid calendar date!
Expiration date 2/30/2011: invalid calendar date!
John Doe added.
John Doe is already in the database.
john doe is already in the database.
ABC: invalid location!
Jane Doe added.
April March added.
Mary Lindsey added.
Duke Ellington added.
Roy Brooks added.
Roy Brooks added.
Kate Lindsey added.
Carl Brown added.
Paul Siegel added.
Bill Scanlan added.

-list of members-

John Doe, DOB: 1/20/2004, Membership expires 3/30/2023, Location: BRIDGEWATER, 08807, SOMERSET
John Doe, DOB: 1/20/2003, Membership expired 3/30/2021, Location: BRIDGEWATER, 08807, SOMERSET
Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
Roy Brooks, DOB: 8/8/1977, Membership expired 9/30/2020, Location: SOMERVILLE, 08876, SOMERSET
Roy Brooks, DOB: 9/9/1977, Membership expires 9/30/2023, Location: SOMERVILLE, 08876, SOMERSET
Kate Lindsey, DOB: 7/15/1977, Membership expires 12/31/2023, Location: SOMERVILLE, 08876, SOMERSET
Carl Brown, DOB: 10/7/1991, Membership expires 3/31/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Paul Siegel, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
-end of list-

-list of members sorted by county and zipcode-

Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
Paul Siegel, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Carl Brown, DOB: 10/7/1991, Membership expires 3/31/2023, Location: PISCATAWAY, 08854, MIDDLESEX
John Doe, DOB: 1/20/2003, Membership expired 3/30/2021, Location: BRIDGEWATER, 08807, SOMERSET
John Doe, DOB: 1/20/2004, Membership expires 3/30/2023, Location: BRIDGEWATER, 08807, SOMERSET
Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
Roy Brooks, DOB: 9/9/1977, Membership expires 9/30/2023, Location: SOMERVILLE, 08876, SOMERSET
Kate Lindsey, DOB: 7/15/1977, Membership expires 12/31/2023, Location: SOMERVILLE, 08876, SOMERSET
Roy Brooks, DOB: 8/8/1977, Membership expired 9/30/2020, Location: SOMERVILLE, 08876, SOMERSET
-end of list-

-list of members sorted by last name, and first name-

Roy Brooks, DOB: 9/9/1977, Membership expires 9/30/2023, Location: SOMERVILLE, 08876, SOMERSET
Roy Brooks, DOB: 8/8/1977, Membership expired 9/30/2020, Location: SOMERVILLE, 08876, SOMERSET
Carl Brown, DOB: 10/7/1991, Membership expires 3/31/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
John Doe, DOB: 1/20/2003, Membership expired 3/30/2021, Location: BRIDGEWATER, 08807, SOMERSET
John Doe, DOB: 1/20/2004, Membership expires 3/30/2023, Location: BRIDGEWATER, 08807, SOMERSET
Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
Kate Lindsey, DOB: 7/15/1977, Membership expires 12/31/2023, Location: SOMERVILLE, 08876, SOMERSET
Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
Paul Siegel, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
-end of list-

-list of members sorted by membership expiration date-

Roy Brooks, DOB: 8/8/1977, Membership expired 9/30/2020, Location: SOMERVILLE, 08876, SOMERSET
John Doe, DOB: 1/20/2003, Membership expired 3/30/2021, Location: BRIDGEWATER, 08807, SOMERSET
Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
Paul Siegel, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
John Doe, DOB: 1/20/2004, Membership expires 3/30/2023, Location: BRIDGEWATER, 08807, SOMERSET
Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
Carl Brown, DOB: 10/7/1991, Membership expires 3/31/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX
Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
Roy Brooks, DOB: 9/9/1977, Membership expires 9/30/2023, Location: SOMERVILLE, 08876, SOMERSET
Kate Lindsey, DOB: 7/15/1977, Membership expires 12/31/2023, Location: SOMERVILLE, 08876, SOMERSET

-end of list-

Paul Siegel removed.

-list of members-

Roy Brooks, DOB: 8/8/1977, Membership expired 9/30/2020, Location: SOMERVILLE, 08876, SOMERSET
 John Doe, DOB: 1/20/2003, Membership expired 3/30/2021, Location: BRIDGEWATER, 08807, SOMERSET
 Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
 John Doe, DOB: 1/20/2004, Membership expires 3/30/2023, Location: BRIDGEWATER, 08807, SOMERSET
 Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
 Carl Brown, DOB: 10/7/1991, Membership expires 3/31/2023, Location: PISCATAWAY, 08854, MIDDLESEX
 Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
 April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX
 Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
 Roy Brooks, DOB: 9/9/1977, Membership expires 9/30/2023, Location: SOMERVILLE, 08876, SOMERSET
 Kate Lindsey, DOB: 7/15/1977, Membership expires 12/31/2023, Location: SOMERVILLE, 08876, SOMERSET
 -end of list-

Paul Siegel is not in the database.

Paul Siegel added.

-list of members sorted by membership expiration date-

Roy Brooks, DOB: 8/8/1977, Membership expired 9/30/2020, Location: SOMERVILLE, 08876, SOMERSET
 John Doe, DOB: 1/20/2003, Membership expired 3/30/2021, Location: BRIDGEWATER, 08807, SOMERSET
 Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX
 John Doe, DOB: 1/20/2004, Membership expires 3/30/2023, Location: BRIDGEWATER, 08807, SOMERSET
 Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
 Carl Brown, DOB: 10/7/1991, Membership expires 3/31/2023, Location: PISCATAWAY, 08854, MIDDLESEX
 Paul Siegel, DOB: 6/30/1999, Membership expires 3/31/2023, Location: FRANKLIN, 08873, SOMERSET
 Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
 April March, DOB: 3/31/1990, Membership expires 6/30/2023, Location: PISCATAWAY, 08854, MIDDLESEX
 Roy Brooks, DOB: 9/9/1977, Membership expires 9/30/2023, Location: SOMERVILLE, 08876, SOMERSET
 Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
 Kate Lindsey, DOB: 7/15/1977, Membership expires 12/31/2023, Location: SOMERVILLE, 08876, SOMERSET
 -end of list-

-Fitness classes-

Pilates – JENNIFER 9:30

Spinning – DENISE 14:00

Cardio – KIM 14:00

DOB 13/8/1977: invalid calendar date!

Roy Brooks 8/8/1977 membership expired.

Mary Lindsey checked in Pilates.

Mary Lindsey has already checked in Pilates.

abc class does not exist.

Jane Doe checked in Pilates.

Mary Lindsey checked in Spinning.

Mary Lindsey has already checked in Spinning.

Cardio time conflict -- Mary Lindsey has already checked in Spinning.

Duke Ellington checked in Cardio.

Duke Ellington has already checked in Cardio.

John Doe 1/20/2003 membership expired.

Bill Scanlan 11/20/2003 is not in the database.

Bill Scanlan checked in Cardio.

Spinning time conflict -- Bill Scanlan has already checked in Cardio.

Paul Siegel checked in Spinning.

-Fitness classes-

Pilates – JENNIFER 9:30

** participants **

Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET

Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
Spinning – DENISE 14:00
** participants **
Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
Paul Siegel, DOB: 6/30/1999, Membership expires 3/31/2023, Location: FRANKLIN, 08873, SOMERSET
Cardio – KIM 14:00
** participants **
Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET
Bill Scanlan, DOB: 5/1/1999, Membership expires 1/31/2023, Location: EDISON, 08837, MIDDLESEX

DOB 12/32/1989: invalid calendar date!
ABC class does not exist.
Mary Lindsey dropped Pilates.
Mary Lindsey is not a participant in Pilates.
Paul Siegel dropped Spinning.
Paul Siegel is not a participant in Spinning.
Bill Scanlan is not a participant in Cardio.
Bill Scanlan dropped Cardio.

-Fitness classes-
Pilates – JENNIFER 9:30
** participants **
Jane Doe, DOB: 5/1/1996, Membership expires 3/30/2023, Location: EDISON, 08837, MIDDLESEX
Spinning – DENISE 14:00
** participants **
Mary Lindsey, DOB: 12/1/1989, Membership expires 5/31/2023, Location: FRANKLIN, 08873, SOMERSET
Cardio – KIM 14:00
** participants **
Duke Ellington, DOB: 2/29/2000, Membership expires 9/30/2023, Location: FRANKLIN, 08873, SOMERSET

q is an invalid command!
Gym Manager terminated.