

Due Date

Monday, September 27, 11:59pm.

Submission

1. One of the team members is responsible to submit **the zipped project folder** to Canvas. The zipped file **MUST** include the following grading items.
 - Source folder **src** [40 points]
 - Must use a package to organize the source files *.java; use lowercase letters for the package name
 - **MUST** include all team members' names using the **@author** tag in the comment block on top of EVERY Java class.
 - Include a testbed main in the **Date class** and implement the test cases in the test design document. [10 points]
 - Javadoc folder **doc**, including all the files generated. [5 points]
 - Test design document. You must use a document editor and include all test cases you used to test the **isValid()** method of the Date class. [10 points]
2. The submission button on Canvas will disappear after **September 27, 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. **DO NOT** wait until the last minute. **DO NOT** send the projects to me or the graders through the emails.

Project Description

You will develop a simple software to manage the information of a collection of albums owned by the user. This project uses the IDE console as the user interface and utilizes the Java standard input and output to manage the read/write of the data being processed.

In this project, you will implement an array-based container class **Collection** and use it to hold a list of albums the user owns. Each album shall include the title, artist, genre, release date, and a flag identifying the availability. An instance of the Collection class is a growable array list data structure with an initial capacity of 4, and automatically grows (increases) the capacity by 4 whenever it is full. The array list does not decrease in capacity. You are **NOT allowed** to use the Java library class **ArrayList**, or you will **get 0 points** for this project.

You must include a **CollectionManger class** as the user interface class to manage the collection. The user shall be able to **add/delete** albums to/from the collection. In addition, the user can set the availability of an album when **lending** out the album to a friend or a friend is **returning** the album. The software shall also provide the functionality of **displaying** the album collection sorted by the release dates or by the genres, in ascending order.

You can use the Java library classes **Scanner** and **StringTokenizer** to read the command lines from the console. When your project starts running, it shall display "Collection Manager starts running.". Next, it will read and process the command lines continuously until the user enters the "Q" command to quit. Before the project stops running, display "Collection Manager terminated."

A command line always begins with a command, in uppercase letters, and followed by a comma and some data tokens. Each data token is also delimited by a comma. Some examples of valid command lines are demonstrated below. All commands are case-sensitive, which means the commands with lowercase letters are invalid. You are required to deal with bad commands that are not supported. Your software must support the following commands.

1. A command, to **add an album** to the collection, for example,
A,Famous Friends,Chris Young,Country,8/6/2021

The above command line adds an album to the collection, including the title, artist, genre, and the release date. If the command line includes a random genre, use “unknown” as the genre. You should set the album as available by default when you are adding a new album to the collection. The release date shall be given in **mm/dd/yyyy** format. You must always check if the date is valid with the `isValid()` method in the `Date` class. You should also check if the album is already in the collection before you add it. For simplicity, we consider two albums as equal if their title and artist are the same. In this case, do not add it to the collection. See sample output for the messages to display.

2. **D** command to **remove an album** from the collection, for example,

```
D,Famous Friends,Chris Young
```

The above command line removes an album from the collection provided the title and artist. Let’s assume that multiple albums may have the same artist, or same title, but the combination of the tile and artist is unique. Only the album with the matching title and artist will be deleted from the collection. See sample output for the messages to display.

3. **L** command to **lend out an album** to a friend, for example

```
L,Famous Friends,Chris Young
```

The above command line sets the album to be “not available”. For simplicity, we do not keep track of the friend’s information. Display proper messages depending on the results. See sample output.

4. **R** command to **return an album** when the friend is returning the album, for example

```
R,Famous Friends,Chris Young
```

The command line above sets the album to be “available”. Display proper messages depending on the results. See sample output.

5. **P** command to display the list of albums in the collection, for example,

```
P    //display the collection without specifying the order
PD   //display the collection sorted by the release dates
PG   //display the collection sorted by the genres
```

6. **Q** command to stop the program execution and display "Collection Manager terminated."

Program Requirement

1. You **MUST** follow the software development ground rules. You will lose points if you are not following the rules.
2. There are sample input and output at the end of this document for your reference. The graders will be using the sample input as the test cases to test your project. Your project should take the sample input in sequence without getting any exception and without terminating abnormally. You will **lose 2 points** for each incorrect output or each exception.
3. Each Java class must go in a separate file. **-2 points** if you put more than one Java class into a file.
4. Your program **MUST** handle bad commands! **-2 points** for each bad command not handled.
5. You are not allowed to use any Java library classes, except **Scanner**, **StringTokenizer** and **Calendar** classes. **You will lose 5 points** for each additional Java library class imported.
6. 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.

7. You **MUST** include the classes below. **-10 points** for each class missing. You **CAN** add necessary constructors, private methods (helper methods), and other public methods to each class. You can also create an additional class to define all the constant identifiers needed in this project.

(a) **Album class**

```
public class Album {
    private String title;
    private String artist;
    private Genre genre; //enum class; Classical, Country, Jazz, Pop, Unknown
    private Date    releaseDate;
    private boolean isAvailable;
    ...
    @Override
    public boolean equals(Object obj) { }
    ...
    @Override
    public String toString() { }
}
```

- This class defines the abstract data type Album, which encapsulates the data fields (instance variables) and methods of an album. Instance variables are considered as “global variables” and are visible to all methods within the class. A good software development practice is to declare a variable locally if it is not required by all methods. 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 an album in the following format.

Tiranno::Kate Lindsey::Classical::5/28/2020::is available

- **equals()** method returns true if the titles and artists are the same for the two albums.
- You **CANNOT** change the signatures of the **toString()** and **equals()** methods. You cannot remove the **@Override** tags. **-2 points** for each violation.

(b) **Collection class**

```
public class Collection {
    private Album[] albums;
    private int numAlbums; //number of albums currently in the collection
    ...
    private int find(Album album) {} //find the album index, or return NOT_FOUND
    private void grow() {} //increase the capacity of the array list by 4
    public boolean add(Album album) {}
    public boolean remove(Album album) {}
    public boolean lendingOut(Album album) {} //set to not available
    public boolean returnAlbum(Album album) {} //set to available
    public void print() {} //display the list without specifying the order
    public void printByReleaseDate() {}
    public void printByGenre() {}
}
```

- This is the **container class** that defines the array list data structure to hold the album collection and provide the operations to manage the collection. You **CANNOT** change or add instance variables. **-2 points** for each violation.

- You must implement the above methods and you CANNOT change the signatures of the methods. **-2 points** for each method not implemented or not used.
- You CANNOT use **System.out** statements in this class, EXCEPT the three **print()** methods, **-2 points** for each violation.
- The **find()** method searches an album in the collection and returns the index if it is found, it returns -1 if it is not in the collection.
- The **remove()** method delete an album from the collection. This method maintains the current sequence of albums in the array after the deletion, **-3 points** if this is not done.
- You can sort the collection using any sorting algorithms. You must write the code for the sorting. You cannot use **Arrays.sort()** or **System.arraycopy()** or you will **lose 5 points**.

(c) Date class

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

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

    ...
    public boolean isValid() {}
    ...
    @Override
    public int compareTo(Date date) {}
}
```

- You CANNOT change or add instance variables, and you CANNOT use **System.out** statements in this class. **-2 points** for each violation.
- You must implement the constructors and methods listed above. You must override the **compareTo()** method of the Java interface Comparable. You will need this method to sort by the release dates.
- The **isValid()** method checks if a date is valid. The collection will not include the albums before the 80's.
 - For a date with the year less than 1980 or a date beyond today's date is invalid.
 - 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 normal 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 identifiers for the constants.

```
public static final int QUADRENNIAL = 4;
public static final int CENTENNIAL = 100;
public static final int QUATERCENTENNIAL = 400;
public static final int THE_EIGHTYS = 1980;
```

- 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. You must write a testbed main and implement the test cases. You must follow the instructions in the "Test Specification" section

in the Software Development Ground Rules. You will **lose 10 points** if you do not submit the test document, or not follow the format. In the testbed main, you **MUST** write code to print out the test results to the console showing the test cases are passed or not passed. **The testbed main is worth 10 points.** Please use “//” comments to identify the test case numbers in the testbed main.

(d) **CollectionManager class**

- This class is the **user interface class** that handles the input (read) and output (write).
- You must define a **public void run()** method that includes a while loop to continuously read the command lines from the console. You will **lose 5 points** if the method is missing. Make sure you keep this method **under 35 lines** for readability, or you will **lose 2 points**.
- You can define necessary instance variables and private methods for handling the commands. Make sure you follow the ground rules and have a good programming style.

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

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

8. You must follow the instructions in the Software Development Ground Rules and comment your code. 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 and include it in your project folder 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 (Test Cases)

```
p
a
pd
pg
l blah blah blah
d
r
AA
PP
RR
P
PD
PG
A,Change,Anika,pop,7/23/2021
A,Change,Anika,pop,7/23/2021
A,Change,April March,pop,7/23/2021
A,Spiral,Darkside,pop,7/1/2021
A,Debussy:La Mer,Leonard Bernstein,classical,2/28/1995
A,Debussy:La mer,Gary Bertini,classical,1/1//2006
A,Heart-Shaped Scars,Dot Allison,pop,7/31/2021
A,Tiranno,Kate Lindsey,classical,5/28/2020
A,Complete Atlantic,Mose Allison,jazz,3/1/2021
A,Never No Lament,Duke Ellington,jazz,4/1/2003
A,Understanding,Roy Brooks,jazz,2/29/2008
A,Change,April March,pop,7/23/2021
D,Change,April March
A,Change,April March,pop,7/23/2021
```

A,Famous Friends,Chris Young,Country,8/6/2021
 A,The Grass Is Blue,Dolly Parton,country,10/26/1999
 A,The Essential Willie Nelson, Willie Nelson,country,4/1/2003
 D,Famous Friends,Chris Young
 P
 D,FAMOUS FRIEND,CHRIS YOUNG
 PD
 PG
 L,Change,April March
 P
 L,Change,April March
 R,Change,April March
 P
 R,Change,April March
 L,Change,xxxx
 R,Change,xxxx
 A,Speak Now,Taylor Swift,xxx,10/25/2010
 P
 A,A fun album,xxx,xxx,31/2/2000
 A,A fun album,xxx,xxx,13/2/2020
 A,A fun album,xxx,xxx,2/29/2021
 A,A fun album,xxx,xxx,2/29/2009
 A,A fun album,xxx,xxx,4/31/2009
 A,A fun album,xxx,xxx,3/32/2009
 A,A fun album,xxx,xxx,3/31/1800
 A,A fun album,xxx,xxx,10/30/2022
 A,A fun album,xxx,xxx,11/30/2021
 A,A fun album,xxx,xxx,11/30/1979
 PG

 q
 Q

A,this line should not be processed,xxx,xxx,11/11/2020

Sample Output

Collection Manager starts running.
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 Invalid command!
 The collection is empty!
 The collection is empty!
 The collection is empty!
 Change::Anika::Pop::7/23/2021::is available >> added.
 Change::Anika::Pop::7/23/2021::is available >> is already in the collection.
 Change::April March::Pop::7/23/2021::is available >> added.
 Spiral::Darkside::Pop::7/1/2021::is available >> added.
 Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available >> added.
 Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available >> added.
 Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available >> added.
 Tiranno::Kate Lindsey::Classical::5/28/2020::is available >> added.
 Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available >> added.
 Never No Lament::Duke Ellington::Jazz::4/1/2003::is available >> added.
 Understanding::Roy Brooks::Jazz::2/29/2008::is available >> added.
 Change::April March::Pop::7/23/2021::is available >> is already in the collection.

Change::April March >> deleted.
 Change::April March::Pop::7/23/2021::is available >> added.
 Famous Friends::Chris Young::Country::8/6/2021::is available >> added.
 The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available >> added.
 The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available >> added.
 Famous Friends::Chris Young >> deleted.
 *List of albums in the collection.
 Change::Anika::Pop::7/23/2021::is available
 Spiral::Darkside::Pop::7/1/2021::is available
 Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
 Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
 Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available
 Tiranno::Kate Lindsey::Classical::5/28/2020::is available
 Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
 Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
 Understanding::Roy Brooks::Jazz::2/29/2008::is available
 Change::April March::Pop::7/23/2021::is available
 The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
 The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
 *End of list
 FAMOUS FRIEND::CHRIS YOUNG >> is not in the collection.
 *Album collection by the release dates.
 Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
 The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
 Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
 The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
 Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
 Understanding::Roy Brooks::Jazz::2/29/2008::is available
 Tiranno::Kate Lindsey::Classical::5/28/2020::is available
 Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
 Spiral::Darkside::Pop::7/1/2021::is available
 Change::April March::Pop::7/23/2021::is available
 Change::Anika::Pop::7/23/2021::is available
 Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available
 *End of list
 *Album collection by genre.
 Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
 Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
 Tiranno::Kate Lindsey::Classical::5/28/2020::is available
 The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
 The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
 Understanding::Roy Brooks::Jazz::2/29/2008::is available
 Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
 Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
 Spiral::Darkside::Pop::7/1/2021::is available
 Change::April March::Pop::7/23/2021::is available
 Change::Anika::Pop::7/23/2021::is available
 Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available
 *End of list
 Change::April March >> lending out and set to not available.
 *List of albums in the collection.
 Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
 Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
 Tiranno::Kate Lindsey::Classical::5/28/2020::is available
 The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
 The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
 Understanding::Roy Brooks::Jazz::2/29/2008::is available
 Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
 Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
 Spiral::Darkside::Pop::7/1/2021::is available
 Change::April March::Pop::7/23/2021::is not available
 Change::Anika::Pop::7/23/2021::is available
 Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available

```

*End of list
Change::April March >> is not available.
Change::April March >> returning and set to available.
*List of albums in the collection.
Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
Tiranno::Kate Lindsey::Classical::5/28/2020::is available
The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
Understanding::Roy Brooks::Jazz::2/29/2008::is available
Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
Spiral::Darkside::Pop::7/1/2021::is available
Change::April March::Pop::7/23/2021::is available
Change::Anika::Pop::7/23/2021::is available
Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available
*End of list
Change::April March >> return cannot be completed.
Change::xxxx >> is not available.
Change::xxxx >> return cannot be completed.
Speak Now::Taylor Swift::Unknown::10/25/2010::is available >> added.
*List of albums in the collection.
Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
Tiranno::Kate Lindsey::Classical::5/28/2020::is available
The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
Understanding::Roy Brooks::Jazz::2/29/2008::is available
Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
Spiral::Darkside::Pop::7/1/2021::is available
Change::April March::Pop::7/23/2021::is available
Change::Anika::Pop::7/23/2021::is available
Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available
Speak Now::Taylor Swift::Unknown::10/25/2010::is available
*End of list
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
Invalid Date!
*Album collection by genre.
Debussy:La Mer::Leonard Bernstein::Classical::2/28/1995::is available
Debussy:La mer::Gary Bertini::Classical::1/1/2006::is available
Tiranno::Kate Lindsey::Classical::5/28/2020::is available
The Essential Willie Nelson:: Willie Nelson::Country::4/1/2003::is available
The Grass Is Blue::Dolly Parton::Country::10/26/1999::is available
Understanding::Roy Brooks::Jazz::2/29/2008::is available
Never No Lament::Duke Ellington::Jazz::4/1/2003::is available
Complete Atlantic::Mose Allison::Jazz::3/1/2021::is available
Spiral::Darkside::Pop::7/1/2021::is available
Change::April March::Pop::7/23/2021::is available
Change::Anika::Pop::7/23/2021::is available
Heart-Shaped Scars::Dot Allison::Pop::7/31/2021::is available
Speak Now::Taylor Swift::Unknown::10/25/2010::is available
*End of list
Invalid command!
Collection Manager terminated.

```