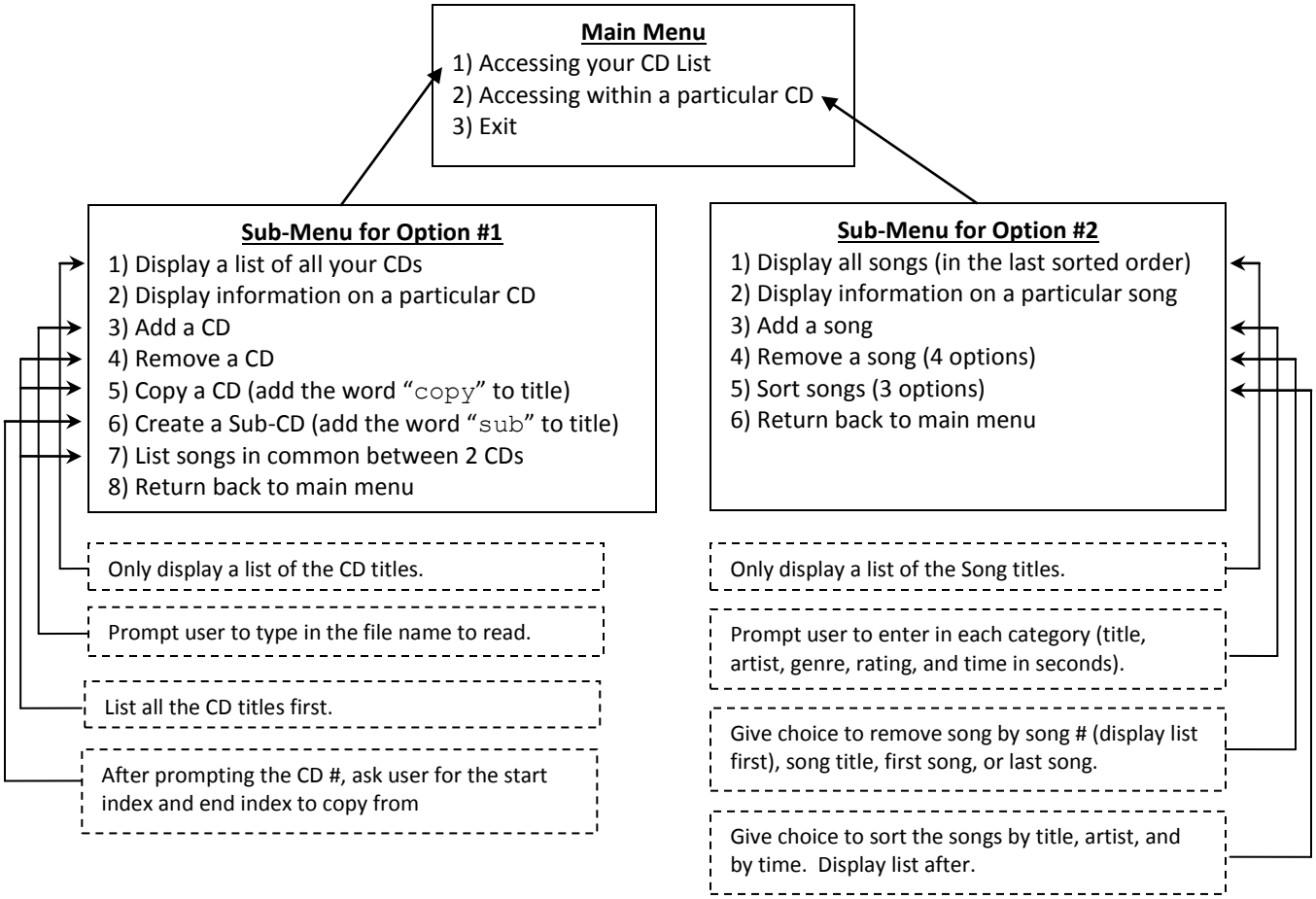
ICS4U1 Date: _____

Assignment #4 – OOP: My CD Collection

For this assignment, you will be creating an interactive program that involves organizing your music CD collection. You will have a list of CDs to keep track of, and each CD has its own title and list of songs.

Your program will contain the following menu system.



You will implement 3 inter-related classes, as well as a class that contains the main program:

1. **`Driver.java`**
   This is the class that contains the main program. It should also contain an `ArrayList` of your CDs.

2. **`CD.java`**
   In this class, you will need to keep track of the CD title, the number of songs in the CD, an `ArrayList` of all the Songs in the CD, and a `Time` variable that contains the total time of all the songs in the CD. *(Constructor: Should have 2 parameters – CD title and number of songs.)*

3. **`Song.java`**
   A `Song` has a title, artist name, genre, rating (out of 5), and a `Time` variable that contains the length of the song. *(Constructor: Should have 5 parameters.)*

4. **`Time.java`**
   A `Time` object has two variables, and that is the time in number of minutes and number of seconds. *(Constructors: You should have 2 different constructors for this class: 1 - with String parameter (for strings in the format of `mm:ss`); 2 - with seconds parameter)*

Here are a few hints/comments to get you started:

- When option #2 is selected from the main menu, you should first ask the user to select which CD. Display the sub-menu after.

- For the options where you list all the CDs/songs first, label each CD/song with numbers starting with 1. The user can type the in the number to make their selections.

- The following is a sample CD input file:

```
Avatar Soundtrack                    ← CD title
2                                    ← Number of songs in CD
Jake enters his avatar world         ← Song #1 title
James Horner                         ← Song #1 artist
Soundtrack                           ← Song #1 genre
4                                    ← Song #1 rating
5:24                                 ← Song #1 length of song
I See You                            ← Song #2 title
Leona Lewis                          …
Soundtrack
5
4:20
```

- All class variables should be `private`. Use "getter" and "setter" methods to access them.

- Binary search should be used when possible, to make your searches more efficient. In other words, you should implement the `Comparable` and `Comparators` interface as needed.

- Duplicate CDs and songs can be added, since it is possible for you to own duplicate CDs, and a CD can have duplicate songs.

- Songs:
  - When removing a song by its title, you must remove ALL instances of that song in the CD (song may be duplicated in CD.)
  - When removing a song, remember to not only remove the song from the list, but also to update the other class variables (# of songs, total time)
  - When a CD is first created, the initial number of songs is set. This can change if new songs are added later on.
  - When making a copy or sub-CD, make sure the new CDs contains the songs, not referring to the original CD. *In other words, if the order of the songs in original CD changes, it should not affect the copied or sub-CD at all!*

- You will notice that most options in sub-menu #1 are done in the `driver` class, while most options in sub-menu #2 are done in the `CD` class. Make methods in these classes to help you reduce duplicate code.

- Like always, your program should perform all error checks. The only part you can assume will always be valid is the format of the CD input files.

- It is up to you to design exactly how this program runs. You want to make it as user-friendly as possible – menu should be easy to access and results should be **clearly** and **neatly** displayed.

ICS4U1 Date: _____

Due Dates:

☆ There will be 3 separate due dates for this assignment:

Due date #1: _____

☺ A memory diagram to show how all the classes relate to each other.
- The diagram must clearly show how all FOUR classes relate.
- For each class/object that you create, you must clearly show all the fields that each contains.
- For each class/object, describe some important methods. (Skip getters and setters.)
  - You can just give the method name – no need to give formal heading.
- You can make up sample data to help you.

☺ To get started, look at my example memory diagram for the `Movies` assignment. Your memory diagram should look similar (ie. arrows, boxes, etc.)

Due date #2: _____

☺ A skeleton of ALL the classes (`CD`, `Song`, `Time`)
- Each class must have all the necessary instance and static variables declared. (You can always ADD more variables later, but you should have the basic ones as defined in the assignment handout.)
- Also define the constructor for each class. Again, you can change these later if you need to.

☺ `Driver` class: You must have a main program that properly reads in options from the user, with proper error checks done.

☺ You must have the following functions properly implemented:
- ✓ Display a list of added CDs *(Menu #1 – Submenu #1)*
- ✓ Be able to add multiple CDs onto `ArrayList` (read in from input file) *(Menu #1 – Submenu #2)*
- ✓ Display a list of songs in a particular CD *(Menu #2 – Submenu #1)*

Due date #3: _____

☺ The rest of your program.

☺ Don't forget the necessary comments:
- Class comments for each class (Name, description)
- Method comments (for each method) – you can group the "getter" and "setter" methods. (Purpose, parameters, return values)
- Internal comments

Final notes:

✸ You are given the starter code for your `Driver.java` class, as well as a sample input file.

✸ Please start on this assignment early and make good use of the time given in class. This assignment does require much more work than the last.

✸ You will definitely gain a much stronger understanding of the object-oriented concepts after completion of this assignment!