Assignment requirements

Prescription Course Management System

This case study is relatively straightforward, and hence can be tackled by most beginning modelers fairly effortlessly.

Background

A School wishes for us to design and develop an automated Course Management System (CM). The requirements are as follows:

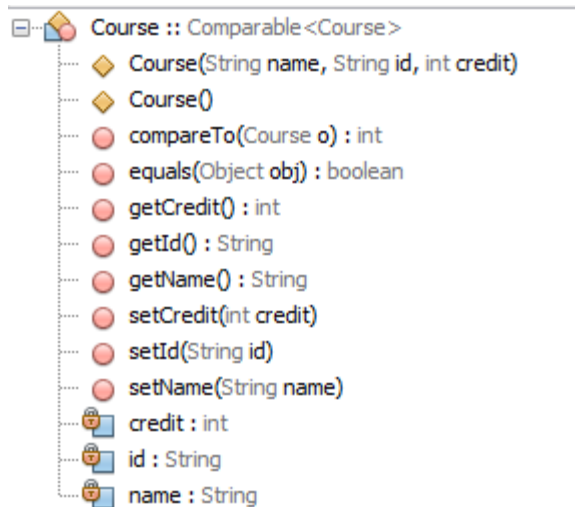The system is to keep track of the following information for each course:

- Course id as a string
- Course name as a string
- Number of credit as a whole number

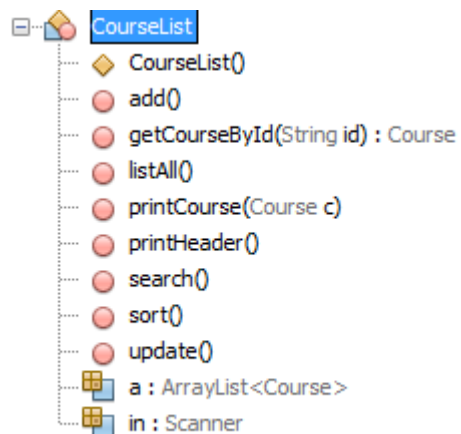The system is required to support the following queries:

1. A list of all available courses in the system
2. Search and display information of a course by course id
3. Record/Add information of course
4. Sort all courses by number of credit as ascending
5. Update information of a specific course (by course id)

You should des:

1. Course class: This will hold the information of course and add some needed operation.

```
⊟┄🔺 Course :: Comparable<Course>
   ┄◇ Course(String name, String id, int credit)
   ┄◇ Course()
   ┄⦿ compareTo(Course o) : int
   ┄⦿ equals(Object obj) : boolean
   ┄⦿ getCredit() : int
   ┄⦿ getId() : String
   ┄⦿ getName() : String
   ┄⦿ setCredit(int credit)
   ┄⦿ setId(String id)
   ┄⦿ setName(String name)
   ┄🔒 credit : int
   ┄🔒 id : String
   ┄🔒 name : String
```

2. CourseList class: This will hold and manage the list of courses, the list of courses can be used as collection.

```
⊟┄🔺 CourseList
   ┄◇ CourseList()
   ┄⦿ add()
   ┄⦿ getCourseById(String id) : Course
   ┄⦿ listAll()
   ┄⦿ printCourse(Course c)
   ┄⦿ printHeader()
   ┄⦿ search()
   ┄⦿ sort()
   ┄⦿ update()
   ┄▦ a : ArrayList<Course>
   ┄▦ in : Scanner
```

3. And finally, make your own main function and test all the above operations

The program output might look something like:

```
COURSE MANAGEMENT SYSTEM
1. A list of all available courses in the system
2. Search and display information of  a course by course id
3. Record/Add information of course
4. Sort all courses by number of credit as ascending
5. Update information of a specific course (by course id)
0. Exit
Select your choice: 3
Enter course id: PRO192
Enter course name: OOP with Java
Enter course credit: 3
Information of course has been added


COURSE MANAGEMENT SYSTEM
1. A list of all available courses in the system
2. Search and display information of  a course by course id
3. Record/Add information of course
4. Sort all courses by number of credit as ascending
5. Update information of a specific course (by course id)
0. Exit
Select your choice: 1
Course id Course name                    Course credit
PRO192     OOP with Java                   3


COURSE MANAGEMENT SYSTEM
1. A list of all available courses in the system
2. Search and display information of  a course by course id
3. Record/Add information of course
4. Sort all courses by number of credit as ascending
5. Update information of a specific course (by course id)
0. Exit
Select your choice: 5
Enter course id: PRO192
Information of course
PRO192     OOP with Java                   3
Enter new course credit: 4
Information of course has been updated

COURSE MANAGEMENT SYSTEM
1. A list of all available courses in the system
2. Search and display information of  a course by course id
3. Record/Add information of course
4. Sort all courses by number of credit as ascending
5. Update information of a specific course (by course id)
0. Exit
Select your choice: 1
Course id Course name                    Course credit
PRO192     OOP with Java                   4
```

```
COURSE MANAGEMENT SYSTEM
1. A list of all available courses in the system
2. Search and display information of  a course by course id
3. Record/Add information of course
4. Sort all courses by number of credit as ascending
5. Update information of a specific course (by course id)
0. Exit
Select your choice: 2
Enter course id: pro192
Information of course
PRO192    OOP with Java                3
```