Assignment questions Mastering java 8

DESIGN PATTERNS and SOLID PRINCIPLES with JAVA

Problem Statement: Singleton Pattern

```
public class Course{
 private Interger courseId;
private String courseName;
private Double courseFee;
private Interger duration;
// getter/ setter & constructors
}
```

Convert the below class CourseData by adding your code in such a manner so that only one object of this class can be possible. Make sure that creation of duplicate object or cloning is not possible.

```
class CourseData{
private static List<Course> courseList=new ArrayList();
static{
 courseList.add(new Course(101,"BTech",450000.00,48));
 courseList.add(new Course(202,"MTech",405000.00,24));
 courseList.add(new Course(303,"BCA",425000.00,48));
 courseList.add(new Course(404,"MCA",450000.00,24));
 }
 // your code
}
```

```java
package designpatternsAndSOLIDprinciples;


//Problem Statement: : Singleton Pattern
//Singleton: Ensures a class has only one instance and provides a global access
point.
public class Course{
     private Integer courseId;
     private String courseName;
     private Double courseFee;
     private Integer duration;
     // getter/ setter & constructors



     public Integer getCourseId() {
             return courseId;
     }
     public Course(Integer courseId, String courseName, Double courseFee,
Integer duration) {
             super();
             this.courseId = courseId;
             this.courseName = courseName;
             this.courseFee = courseFee;
             this.duration = duration;
     }
     public Course() {
     }
     public void setCourseId(Integer courseId) {
             this.courseId = courseId;
     }
     public String getCourseName() {
             return courseName;
     }
     public void setCourseName(String courseName) {
             this.courseName = courseName;
     }
     public Double getCourseFee() {
             return courseFee;
     }
     public void setCourseFee(Double courseFee) {
             this.courseFee = courseFee;
     }
     public Integer getDuration() {
             return duration;
     }
     public void setDuration(Integer duration) {
             this.duration = duration;
     }
     @Override
     public String toString() {
             return "Course [courseId=" + courseId + ", courseName=" + courseName
+ ", courseFee=" + courseFee
                          + ", duration=" + duration + "]";
     }

}
```

```java
package designpatternsAndSOLIDprinciples;

import java.util.ArrayList;
import java.util.List;

//singleton design pattern
//Singleton: Ensures a class has only one instance and provides a global access
point.

//Convert the below class CourseData by adding your code in such a manner so that
//only one object of this class can be possible. Make sure that creation of
duplicate
//object or cloning is not possible.

public class CourseData{
        private static List<Course> courseList = new ArrayList<>();
        static{
                courseList.add(new Course(101,"BTech",450000.00,48));
                courseList.add(new Course(202,"MTech",405000.00,24));
                courseList.add(new Course(303,"BCA",425000.00,48));
                courseList.add(new Course(404,"MCA",450000.00,24));
        }

                public void addNewCourse(Course course) {
                        courseList.add(course);
                        }

                public void getAllCourses() {
                        courseList.forEach((o)->System.out.println(o.toString()));

                }

                public void deleteCourse(int courseID) {
                        int index=getCourseIndex(courseID);
                        if(index==-1) {
                                System.out.println("\ncourse not found with given
course id " + courseID);
                        }
                        else
                        courseList.remove(index);
                        System.out.println("\ncourse with id " + courseID+" deleted
success");
                }

                int getCourseIndex(int courseId) {
                        int id=-1;
                        int temp=-1;
                        for(Course c :courseList) {
                                id++;
                                if(c.getCourseId().equals(courseId)) {
                                        temp=id;
                                        break;
                                }
                        }
                        return temp;
                }

                public void updateCourse(Course course) {
                        int id=getCourseIndex(course.getCourseId());
```

```java
                    System.out.println(id);
                    if(id == -1)
                            System.out.println("course with the id not found");
                    else {
                            Course oldValues =  courseList.get(id);
                            int cid = course.getCourseId();
                            String cName = course.getCourseName() == null ? 
oldValues.getCourseName():course.getCourseName() ;
                            double cFee = course.getCourseFee() == null ? 
oldValues.getCourseFee():course.getCourseFee() ;
                            int cDuration = course.getDuration() == null ? 
oldValues.getDuration():course.getDuration() ;

                            courseList.set(id,new 
Course(cid,cName,cFee,cDuration));
                    }

            }
            private static final  CourseData INSTANCE=new CourseData();;
            public  static CourseData getInstance() {
                    return INSTANCE;
            }

            public static void main(String[] args) {
                    CourseData data = CourseData.getInstance();
                    Course course = new Course();

                    course.setCourseId(4000);

                    data.updateCourse(course);
                    data.getAllCourses();
                    data.deleteCourse(400000);

                    System.out.printf("\t After Deleting\n");
                    data.getAllCourses();

                    Course updateCourse = new Course();

                    updateCourse.setCourseId(101);
                    updateCourse.setCourseFee(20000.00);
                    System.out.println("update course with id 101");
                    data.updateCourse(updateCourse);

                    data.getAllCourses();
            }

}
```