Set Interface

Exercise

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

class Student {

    private int studentId;
    private String studentName;
    private int courseId;

    public Student(int studentId, String studentName, int courseId) {
        this.studentId = studentId;
        this.studentName = studentName;
        this.courseId = courseId;
    }

    public int getStudentId() {
        return studentId;
    }

    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }
```

```
public String getStudentName() {
    return studentName;
}

public void setStudentName(String studentName) {
    this.studentName = studentName;
}

public int getCourseId() {
    return courseId;
}

public void setCourseId(int courseId) {
    this.courseId = courseId;
}

@Override
public boolean equals(Object student) {
    if (this == student) return true;
    if (student == null || getClass() != student.getClass()) return false;
    Student otherStudent = (Student) student;
    return studentId == otherStudent.studentId;
}

@Override
public int hashCode() {
    return studentId;
```

```
    }

    @Override
    public String toString() {
        return "Student Id: " + studentId + ", Student Name: " + studentName;
    }
}

class Tester {

    public static Set<Student> findDuplicateEntries(List<Student> students) {
        Set<Student> seenStudents = new HashSet<>();
        Set<Student> duplicateStudents = new HashSet<>();

        for (Student student : students) {
            if (!seenStudents.add(student)) {
                duplicateStudents.add(student);
            }
        }

        return duplicateStudents;
    }

    public static void main(String[] args) {
        List<Student> students = new ArrayList<>();

        students.add(new Student(1001, "Dean", 111));
```

```
        students.add(new Student(1002, "Harley", 112));
        students.add(new Student(1003, "Franklin", 113));
        students.add(new Student(1005, "Arden", 113));
        students.add(new Student(1100, "Juliet", 112));
        students.add(new Student(1003, "Franklin", 111));
        students.add(new Student(1001, "Dean", 114));

        Set<Student> duplicateStudents = findDuplicateEntries(students);

        System.out.println("Students who have applied for re-evaluation in more than one subject:");
        for (Student student : duplicateStudents) {
            System.out.println(student);
        }
    }
}
```

Output-