1. **Implement a Student Class:**

Create a **Student** class with attributes such as **id** (int), **name** (String), and **grade** (double). Include appropriate constructors and getter/setter methods.

1. **Create a Collection for Students:**

Implement a class named **StudentCollectionManager**.

Inside this class, create an ArrayList to store instances of the Student class.

Add methods to:

Add a new student to the collection.

Remove a student by ID.

Display the details of all students in the collection.

Find and display the student with the highest grade.

1. **Use Set for Unique Courses:**

Extend the Student class to include a **Set<String>** attribute named courses representing the courses a student is enrolled in.

Update the **StudentCollectionManager** class to manage unique courses across all students. Ensure that duplicate courses are not added.

1. **Implement a Map for Student Enrollment:**

Modify the StudentCollectionManager class to use a **Map<Integer, Set<String>>** where the key is the student ID and the value is the set of courses that the student is enrolled in.

Testing:

In the main method of a separate Main class:

Create instances of the Student class.

Add students to the StudentCollectionManager and test all implemented methods.

Display the details of all students.

Test the removal of a student by ID.

Display the updated list of students.

Test finding and displaying the student with the highest grade.

Test adding courses to students and ensuring uniqueness.

Display the final enrollment map.

=============================================================================

Second question, you have below structure.

Class Company {

Int id,

String name

}

Class Manager {

{

Int id,

String name,

String companyName,

}

Class Employee {

Int id,

String name,

String managerId

}

Save data in collection as mentioned below.

Company name -> manager name -> employee list

**Note :-** there will be only one manager per company.

**Hint :-** you will have to use Hashmap.

**Third, a. write difference between arraylist and linkedlist.**

**b. write difference between set and list.**

**Fourth,**

1. **Implement an Actor Class:**

Create an Actor class with attributes such as actorId (int), actorName (String), birthDate (String), and nationality (String). Include appropriate constructors and getter/setter methods.

1. **Implement a Movie Class:**

Create a Movie class with attributes such as movieId (int), movieTitle (String), releaseDate (String), and genre (String). Include appropriate constructors and getter/setter methods.

ArrayList of Movies and Actors:

1. **Implement a class named MovieManager.**

Inside this class, use two ArrayLists, one for Movie instances and one for Actor instances.

Add methods to:

Add a new movie to the movie list.

Add a new actor to the actor list.

Remove a movie by movieId.

Remove an actor by actorId.

Display the details of all movies and actors in the lists.

1. **HashMap for Actor-Movie Relationships:**

Modify the MovieManager class to use a HashMap<Integer, ArrayList<Integer>> where the key is the actorId and the value is a list of movieIds associated with that actor.

Add methods to:

Add a movie to an actor's filmography.

Remove a movie from an actor's filmography.

Display the filmography of a specific actor.

1. **Testing:**

In the main method of a separate Main class:

Create instances of the Actor and Movie classes.

Add actors and movies to the MovieManager using the ArrayList and HashMap.

Display the details of all actors and movies.

Add movies to an actor's filmography and display the updated filmography.

Remove movies and actors by their respective IDs and display the updated lists.

1. **Bonus Task: Sorting:**

Implement a method in the MovieManager class to sort movies based on their release dates.

Display the sorted list of movies.

Submission Guidelines:

Submit your Java source code files.

Provide a brief explanation or comments for complex parts of the code.