Mini Project: Tutorial Comment and Feedback Management System in Core Java using only arrays (no collections, no database).

1. **Problem statement**
2. **Class design** (ASCII UML)
3. **Java source code** (fully commented)
4. **All possible test cases** in a table format

---

1. Problem Statement

You are tasked with building a console-based **Tutorial Comment & Feedback Management System**.
The system will manage:

- Tutorials (each with ID, title, description)
- Comments for each tutorial (comment text, username, rating)

Constraints:

- Data must be stored in **arrays only**
- Maximum **100 tutorials** and **10 comments per tutorial**
- Provide features to add tutorials, add comments, view tutorials with comments, search by tutorial ID, update comments, and delete comments.

---

2. Class Design (ASCII UML)

pgsql

CopyEdit

```
+------------------+
|    Comment       |
+------------------+
| String username  |
| String text      |
| int rating       |
+-------------------+
| display()         |
+-------------------+

+----------------------------------+
|           Tutorial               |
+----------------------------------+
| int tutorialId                   |
| String title                     |
| String description               |
| Comment[] comments               |
| int commentCount                 |
+----------------------------------+
| addComment(Comment c)            |
| viewComments()                   |
| updateComment(int index, String) |
| deleteComment(int index)         |
+----------------------------------+

+-------------------------------------+
| TutorialCommentFeedbackSystem       |
+-------------------------------------+
| Tutorial[] tutorials                |
| int tutorialCount                   |
+-------------------------------------+
| addTutorial()                       |
| addCommentToTutorial()              |
| viewAllTutorials()                  |
| searchTutorialById()                |
| updateCommentInTutorial()            |
| deleteCommentFromTutorial()          |
+--------------------------------------+
```

---

3. Java Source Code

```
import java.util.Scanner;

// Represents a comment on a tutorial
class Comment {
    String username;
    String text;
    int rating; // 1 to 5

    public Comment(String username, String text, int rating) {
        this.username = username;
        this.text = text;
        this.rating = rating;
    }

    public void display() {
        System.out.println("  User: " + username);
        System.out.println("  Comment: " + text);
        System.out.println("  Rating: " + rating + "/5");
    }
}

// Represents a tutorial with multiple comments
class Tutorial {
    int tutorialId;
    String title;
    String description;
    Comment[] comments = new Comment[10];
    int commentCount = 0;

    public Tutorial(int tutorialId, String title, String description) {
        this.tutorialId = tutorialId;
        this.title = title;
        this.description = description;
    }

    // Add a comment to this tutorial
    public void addComment(Comment c) {
        if (commentCount < comments.length) {
            comments[commentCount++] = c;
            System.out.println("Comment added successfully!");
        } else {
            System.out.println("Cannot add more comments. Limit
reached!");
        }
    }

    // Display all comments
    public void viewComments() {
        if (commentCount == 0) {
            System.out.println("No comments yet.");
            return;
        }
        for (int i = 0; i < commentCount; i++) {
            System.out.println("Comment #" + (i + 1) + ":");
            comments[i].display();
            System.out.println("-----------------------");
        }
    }

    // Update a specific comment
    public void updateComment(int index, String newText, int newRating) {
        if (index >= 0 && index < commentCount) {
            comments[index].text = newText;
            comments[index].rating = newRating;
            System.out.println("Comment updated successfully!");
        } else {
            System.out.println("Invalid comment index!");
        }
    }

    // Delete a specific comment
    public void deleteComment(int index) {
        if (index >= 0 && index < commentCount) {
            for (int i = index; i < commentCount - 1; i++) {
                comments[i] = comments[i + 1];
            }
            comments[--commentCount] = null;
            System.out.println("Comment deleted successfully!");
        } else {
            System.out.println("Invalid comment index!");
        }
    }
    // Display tutorial details
    public void displayTutorial() {
        System.out.println("Tutorial ID: " + tutorialId);
        System.out.println("Title: " + title);
        System.out.println("Description: " + description);
        viewComments();
    }
}

// Main management system
public class TutorialCommentFeedbackSystem {
    static Tutorial[] tutorials = new Tutorial[100];
    static int tutorialCount = 0;
    static Scanner sc = new Scanner(System.in);

    public static void addTutorial() {
        System.out.print("Enter Tutorial ID: ");
        int id = sc.nextInt();
        sc.nextLine(); // consume newline
        System.out.print("Enter Tutorial Title: ");
        String title = sc.nextLine();
        System.out.print("Enter Tutorial Description: ");
        String desc = sc.nextLine();

        tutorials[tutorialCount++] = new Tutorial(id, title, desc);
        System.out.println("Tutorial added successfully!");
    }
```

```
public static void addCommentToTutorial() {
    System.out.print("Enter Tutorial ID to comment on: ");
    int id = sc.nextInt();
    sc.nextLine();
    Tutorial t = findTutorialById(id);
    if (t != null) {
        System.out.print("Enter Username: ");
        String user = sc.nextLine();
        System.out.print("Enter Comment: ");
        String comment = sc.nextLine();
        System.out.print("Enter Rating (1-5): ");
        int rating = sc.nextInt();
        sc.nextLine();

        t.addComment(new Comment(user, comment, rating));
    } else {
        System.out.println("Tutorial not found!");
    }
}

public static void viewAllTutorials() {
    if (tutorialCount == 0) {
        System.out.println("No tutorials available.");
        return;
    }
    for (int i = 0; i < tutorialCount; i++) {
        tutorials[i].displayTutorial();
        System.out.println("=========================");
    }
```

```java
}

public static void searchTutorialById() {
    System.out.print("Enter Tutorial ID: ");
    int id = sc.nextInt();
    sc.nextLine();
    Tutorial t = findTutorialById(id);
    if (t != null) {
        t.displayTutorial();
    } else {
        System.out.println("Tutorial not found!");
    }
}

public static void updateCommentInTutorial() {
    System.out.print("Enter Tutorial ID: ");
    int id = sc.nextInt();
    sc.nextLine();
    Tutorial t = findTutorialById(id);
    if (t != null) {
        System.out.print("Enter Comment Index (starting from 1): ");
        int index = sc.nextInt() - 1;
        sc.nextLine();
        System.out.print("Enter New Comment: ");
        String newComment = sc.nextLine();
        System.out.print("Enter New Rating: ");
        int newRating = sc.nextInt();
        sc.nextLine();
        t.updateComment(index, newComment, newRating);
    } else {
        System.out.println("Tutorial not found!");
    }
}

public static void deleteCommentFromTutorial() {
    System.out.print("Enter Tutorial ID: ");
    int id = sc.nextInt();
    sc.nextLine();
    Tutorial t = findTutorialById(id);
    if (t != null) {
        System.out.print("Enter Comment Index (starting from 1): ");
        int index = sc.nextInt() - 1;
        sc.nextLine();
        t.deleteComment(index);
    } else {
        System.out.println("Tutorial not found!");
    }
}

private static Tutorial findTutorialById(int id) {
    for (int i = 0; i < tutorialCount; i++) {
        if (tutorials[i].tutorialId == id) {
            return tutorials[i];
        }
    }
    return null;
}
```

```
    public static void main(String[] args) {
        while (true) {
            System.out.println("\n=== Tutorial Comment & Feedback System ===");
            System.out.println("1. Add Tutorial");
            System.out.println("2. Add Comment to Tutorial");
            System.out.println("3. View All Tutorials");
            System.out.println("4. Search Tutorial by ID");
            System.out.println("5. Update Comment in Tutorial");
            System.out.println("6. Delete Comment from Tutorial");
            System.out.println("7. Exit");
            System.out.print("Enter choice: ");
            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1: addTutorial(); break;
                case 2: addCommentToTutorial(); break;
                case 3: viewAllTutorials(); break;
                case 4: searchTutorialById(); break;
                case 5: updateCommentInTutorial(); break;
                case 6: deleteCommentFromTutorial(); break;
                case 7: System.out.println("Exiting..."); return;
                default: System.out.println("Invalid choice!");
            }
        }
    }
}
```

---

4. Possible Test Cases

Test Case ID	Action	Input	Expected Output
TC1	Add Tutorial	ID=101, Title="Java Basics", Desc="Intro to Java"	Tutorial added successfully
TC2	Add Comment	Tutorial ID=101, User="Alice", Text="Great tutorial", Rating=5	Comment added successfully
TC3	View All Tutorials	—	Displays all tutorials with comments
TC4	Search Tutorial	ID=101	Shows tutorial 101 details
TC5	Update Comment	Tutorial ID=101, Comment Index=1, New Text="Very helpful", Rating=4	Comment updated successfully
TC6	Delete Comment	Tutorial ID=101, Comment Index=1	Comment deleted successfully
TC7	Add Comment beyond limit	More than 10 comments for same tutorial	Shows "Cannot add more comments"
TC8	Search non-existing tutorial	ID=999	Shows "Tutorial not found"
TC9	Delete non-existing comment	Comment index > count	Shows "Invalid comment index"
TC10	View tutorials when none exist	—	Shows "No tutorials available"