EduTrack – A Student Performance Monitoring System

You are hired by a startup developing an academic tracking tool named **EduTrack**, which monitors and manages student performance across different courses. The application is designed to count how many student records have been created, maintain a list of grading utilities like GPA calculation and provide global constants like the **maximum GPA** or **passing grade** that do not change. As the lead Java developer, you are tasked with designing the **Student** and **GradeUtils** classes in a way that uses **static keyword effectively**

1. **Use Case 1: Counter for Number of Students (Static Variable)**
    - Every time a new Student object is created, the system should increment a static counter studentCount.
    - The counter should be accessible without creating an object of the class.
2. **Use Case 2: Utility Methods (Static Method)**
    - You need to provide a utility method calculateGPA() that calculates a student's GPA based on an array of marks.
    - This method should be placed in a separate utility class and be accessible globally without instantiating the class.
3. **Use Case 3: Application Constants (Static Final Constants)**
    - Define constants like MAX_GPA = 4.0 and PASS_GRADE = 35 that are shared across the system.
    - These constants should be unchangeable and referenced throughout the app.

Design a small-scale Java application that includes:

- A Student class with instance fields like name, studentId, marks[], and a static variable studentCount.
- A static block to initialize static variables if needed.
- A GradeUtils class with static methods such as calculateGPA() and static constants like MAX_GPA and PASS_GRADE.
- Demonstrate usage of static methods, static variables, and constants inside a main() method or test class.