**Case Study: School Management System**

**Context:**

You are required to develop a console-based School Management System using Core Java. The application will allow users to manage student records and perform various business operations related to school management. The application will utilize Java Collections for storing student records, implement Object-Oriented Programming principles, and include exception handling to manage potential errors.

**Requirements:**

**Features:**

1. **Calculate Average Grade:**
   * Calculate and display the average grade for each student based on their subjects.
2. **Find Top and Bottom Performer:**
   * Determine and display the student(s) with the highest and lowest average grade in the school.
3. **Subject-wise Average Grade:**
   * Calculate and display the average grade for each subject across all students.
4. **Student Attendance Tracking:**
   * Track and display the attendance of each student.
   * Calculate and display the attendance percentage for each student.

**Classes and Objects:**

1. **Student Class:**
   * Attributes: id, name, grades (a map of subjects and their respective grades), attendance (total attended days, total school days).
   * Methods: Constructors, getters and setters, toString method.
2. **Subject Class:**
   * Attributes: subjectId, subjectName.
   * Methods: Constructors, getters and setters, toString method.
3. **SchoolManager Class:**
   * Attributes: HashMap<Integer, Student> students, HashMap<Integer, Subject> subjects.
   * Methods:
     + calculateAverageGrade(int studentId): Calculates the average grade of a student.
     + findTopAndBottomPerformer(): Finds the students with the highest and lowest average grades.
     + calculateSubjectWiseAverageGrade(int subjectId): Calculates the average grade for each subject.
     + trackStudentAttendance(int studentId, int attendedDays, int totalDays): Tracks and calculates the attendance percentage for a student.
     + Helper methods for input validation and exception handling.

**Deliverables:**

1. Complete source code for the School Management System.
2. Documentation including:
   * How to run the application.
   * Instructions for each feature.
   * Explanation of the exception handling implemented.
3. A brief report on the application design and how Object-Oriented principles were applied.