Advanced Computer Programming

Quiz # 1

(September 6, 2024)

You are tasked with creating a program that manages the grades of students for multiple subjects. The program will allow the user to input student information (names, IDs, and grades for 5 subjects), calculate averages, and display statistics about the students' performance.

Requirements:

1. **Student Information**:
    - You will input data for up to 10 students.
    - Each student has a unique ID, a name, and grades for 5 subjects.
    - The program should store the students' names and IDs in two separate 1D arrays and store the grades for each subject in five separate 1D arrays (one array per subject).
2. **Basic Operations**:
    - Input student data: student ID, name, and grades for each subject.
    - Calculate the average grade for each student across all subjects.
    - Display the average, highest, and lowest grade for each subject.
    - Display the overall class average (i.e., the average grade across all students and subjects).
    - Allow the user to display the grades and calculated averages for any specific student.
3. **Edge Cases**:
    - Handle cases where no students have been entered when trying to calculate statistics.
    - Handle invalid input for grades (grades must be between 0 and 100).

Functions to Implement:

1. `inputStudentData()`: Input names, IDs, and grades for multiple students.
2. `calculateStudentAverages()`: Calculate and display the average grade for each student across all subjects.
3. `displaySubjectStatistics()`: Display the average, highest, and lowest grade for each subject.
4. `displayStudentGrades()`: Display the grades and calculated average for a specific student.
5. `calculateOverallAverage()`: Calculate and display the overall class average.

Input and Output:

- The program should allow the user to input data for up to 10 students and display statistics such as individual grades, averages, and subject performance.

Example Input:

For **3 Students**:

- **Student 1**:
  - ID: 101
  - Name: Alice
  - Grades: 85 (Math), 90 (Science), 78 (English), 92 (History), 88 (Art)
- **Student 2**:
  - ID: 102
  - Name: Bob
  - Grades: 75 (Math), 80 (Science), 85 (English), 70 (History), 68 (Art)
- **Student 3**:
  - ID: 103
  - Name: Carol
  - Grades: 90 (Math), 88 (Science), 91 (English), 89 (History), 87 (Art)

Example Output:

1. **Menu**:

```
--- Student Grade Analysis ---
1. Input Student Data
2. Display Averages for Each Student
3. Display Subject Statistics
4. Display Student Grades
5. Display Overall Class Average
6. Exit
Enter your choice: 3
```

2. **Subject Statistics**:

```
Subject: Math
Average: 83.3
Highest Grade: 90
Lowest Grade: 75

Subject: Science
Average: 86.0
Highest Grade: 90
Lowest Grade: 80

Subject: English
Average: 84.6
Highest Grade: 91
Lowest Grade: 78
```

3. **Student Grades**:

```
Enter Student ID: 101
Student: Alice
Math: 85, Science: 90, English: 78, History: 92, Art: 88
Average Grade: 86.6
```

4. **Overall Class Average**:

```
Overall Class Average: 81.7
```

Solution Steps:

1. **Declare Arrays**:
   - One array for storing student names (`String[] names = new String[10]`).
   - One array for storing student IDs (`int[] ids = new int[10]`).
   - Five separate arrays for storing grades for each subject (e.g., `int[] mathGrades = new int[10]`, `int[] scienceGrades = new int[10]`, etc.).
2. **Input Data**:
   - Use a loop to input student data. For each student, prompt the user for the student's name, ID, and grades for the five subjects.
   - Ensure that input validation is done to ensure that all grades fall between 0 and 100.
3. **Calculate Averages**:
   - For each student, use a loop to calculate the average grade across all five subjects.
   - Store and display the calculated averages.
4. **Display Subject Statistics**:
   - For each subject (e.g., Math, Science), calculate the average, highest, and lowest grade by iterating over the corresponding subject array.
   - Display these statistics for all subjects.
5. **Display Individual Grades**:
   - Allow the user to input a student ID to display that student's grades and average.
6. **Calculate Overall Class Average**:
   - Calculate the overall class average by averaging all grades in all subjects across all students.

Expected Features:

- **Input Validation**: Ensure that grades entered are between 0 and 100.
- **Calculation Efficiency**: Use loops to process arrays and calculate the required statistics.
- **Clear Output**: Ensure that the statistics and individual student grades are displayed in a readable and understandable format.