Assignment 1: Student Grade Management System

Objective:

The goal of this assignment is to practice using arrays, if statements, loops, and classes in Java. You will create a simple Student Grade Management System that allows a teacher to input student scores, calculate averages, determine letter grades, and print reports.

Problem Description:

1. **Create a Student class**:
    - The Student class should have the following attributes:
        - name (String) – Name of the student.
        - scores (array of doubles) – An array that stores the scores for each student.
        - average (double) – The average score of the student.
        - letterGrade (char) – The letter grade of the student based on their average score.
2. **Create a GradeBook class**:
    - The GradeBook class should contain the main method and handle input/output that
        - Create an array of Student objects.
        - Allow the teacher to input the number of students.
        - For each student:
            - Input their name.
            - Input the number of scores they have (between 3 and 5).
            - Input each score and store it in the scores array.
            - Calculate their average score.
            - Determine their letter grade (A, B, C, D, F) based on the following scale:
                - A: 90-100
                - B: 80-89
                - C: 70-79
                - D: 60-69
                - F: 0-59
        - Print the student name, average score, and letter grade.

3. **Array Implementation**:
   Use arrays to store:
   - The student names.
   - The scores for each student.
4. **Control Flow**:
   - Use if statements to determine the letter grade.
   - Use loops to process each student and their scores.

Requirements:

- The program should allow for a dynamic number of students (at least 1, no more than 50).
- The program should be robust, handling input correctly (e.g., ensuring scores are within valid ranges).
- The program should display output in a user-friendly format.

Sample Output

```
Enter the number of students: 3

Enter the name of student 1: John

Enter the number of scores for John: 4

Enter score 1: 90

Enter score 2: 85

Enter score 3: 92

Enter score 4: 88


Enter the name of student 2: Alice

Enter the number of scores for Alice: 3

Enter score 1: 78

Enter score 2: 80

Enter score 3: 75
```

```
Enter the name of student 3: Bob
Enter the number of scores for Bob: 5
Enter score 1: 50
Enter score 2: 65
Enter score 3: 55
Enter score 4: 60
Enter score 5: 58

Student: John, Average: 88.75, Grade: B
Student: Alice, Average: 77.67, Grade: C
Student: Bob, Average: 57.6, Grade: F
```

Rubric

Criteria	Points
Student Class (Correct attributes and methods for student data)	20
GradeBook Class (Handles input, output, and logic correctly)	30
Arrays (Correct use of arrays to store student scores)	20
Loops (Correct use of loops to process input for each student)	10
If Statements (Correctly calculates letter grade using if conditions)	10
Code Quality (Clean, readable code with appropriate comments)	10
Total	100