PSEUDOCODE FOR THE GRADEBOOK APP SCRIPT

Purpose:

Calculate a student's grade, determine whether they passed, and generate a summary message with the class average.

Function 1: `getAverage(scores)`

1. **Input**: An array `scores` containing numerical scores.
2. **Initialize Variables**:
    - Set `sum` to `0` to hold the total of all scores.
3. **Process**:
    - Use a `for` loop to iterate over each score in the `scores` array.
        - Add the current score to the `sum`.
4. **Calculate Average**:
    - Divide the `sum` by the length of the `scores` array.
5. **Output**: Return the calculated average.

Function 2: `getGrade(score)`

1. **Input**: A single numerical score.
2. **Process**:
    - Check if the score is exactly `100`.
        - If true, return `"A++"` as the grade.
    - Else, check if the score is greater than or equal to `90`.

        - If true, return "A".
    - Else, check if the score is greater than or equal to 80.
        - If true, return "B".
    - Else, check if the score is greater than or equal to 70.
        - If true, return "C".
    - Else, check if the score is greater than or equal to 60.
        - If true, return "D".
    - Else, return "F" for all other scores below 60.
3. **Output**: Return the corresponding grade as a string.

Function 3: hasPassingGrade(score)

1. **Input**: A single numerical score.
2. **Process**:
    - Call the getGrade function with the given score.
    - Check if the returned grade is not "F".
3. **Output**: Return true if the grade is passing; otherwise, return false.

Function 4: studentMsg(totalScores, studentScore)

1. **Input**:
    - An array totalScores containing the class scores.
    - A single numerical studentScore for the specific student.
2. **Process**:
    - Call the getAverage function with totalScores to calculate the class average.
        - Store the result in classAverage.

- Call the `getGrade` function with `studentScore` to determine the student's grade.
    - Store the result in `studentGrade`.
- Call the `hasPassingGrade` function with `studentScore` to check if the student passed.
    - Store the result in `passGrade`.
- Check if `passGrade` is `true`:
    - Construct a message indicating the class average, the student's grade, and that they passed the course.
- Otherwise:
    - Construct a message indicating the class average, the student's grade, and that they failed the course.

3. **Output**: Return the constructed message.

Main Execution

1. **Call `studentMsg`**:
    - Pass an array of sample scores `[92, 88, 12, 77, 57, 100, 67, 38, 97, 89]` as the class scores.
    - Pass the specific student's score `37`.
2. **Output the Result**:
    - Log the returned message from `studentMsg` to the console using `console.log()`.