Based on the chapters you've shared (Chapters 1–10), here's a **focused and practical programming task** that revises key concepts like **String manipulation**, **arrays**, **methods**, **Scanner input**, **StringBuilder**, and **basic OOP**—without being too trivial or time-consuming.

🧠 Task: Student Grade Analyzer

Objective:
Write a Java program that reads a list of student names and their grades, then performs the following operations:

1. **Input**:
    - Ask the user to enter the number of students.
    - For each student, enter their name and grade (as a String and an integer).
2. **Processing**:
    - Store the data in two arrays: `String[] names` and `int[] grades`.
    - Use `StringBuilder` to build a formatted output string.
    - Compute:
        - The average grade.
        - The highest and lowest grades.
        - The number of students who passed (grade ≥ 60).
3. **Output**:
    - Print the following using `System.out.println` and `String.format`:
        - A table of all students and their grades.
        - The average, max, and min grades.
        - The number of students who passed.

☑ Requirements (to ensure you use built-in methods):

- Use `Scanner` for input.
- Use `StringBuilder` to build the output table.
- Use `Arrays.sort()` to sort the grades (optional, but good practice).
- Use `String.format()` for aligned output.
- Use a helper method to compute the average.

Example Output:

```
Enter number of students: 3
Enter name and grade for student 1: Ali 85
Enter name and grade for student 2: Sara 92
Enter name and grade for student 3: Ahmed 55

=== Grade Report ===
Name    Grade
-------------
Ali     85
Sara    92
Ahmed   55

Average Grade: 77.33
Highest Grade: 92
Lowest Grade: 55
Passed: 2 students
```

Tips:

- Use `Scanner.nextLine()` and `split(" ")` to read name and grade.
- Use `Integer.parseInt()` to convert the grade string to an integer.
- Use `StringBuilder` to append each row of the table.
- Use a loop to compute sum, max, and min.

This task should take **30–60 minutes**, covers **Chapters 2–7 and 10**, and uses many built-in methods and concepts like arrays, strings, scanning, formatting, and simple logic. Let me know if you want a starter code snippet or hints!