Goals & Objectives

The goal of this program is to accept inputs from a user to identify a number of students and subsequent scores associated with each student. Program will find the best score and output the individual scores with grades.

Functional Requirements

1. Prompt user input for number of students
2. Prompt user to enter scores for each student
3. Store each score in Double array
4. Pass scores to method findBestGrade
5. Iterate through array and determine best score
6. Output each student’s score and grade

Pseudocode

Import Scanner Utility

Function Main {

Declare input as New Scanner

Output “Enter the number of students “

Declare numberOfStudents as int

Declare best as double

Declare scores as double array

Output “Enter x scores:”

Add user input to scores in FOR loop

Close input

Assign best as output from method findBestGrade

Iterate through scores with for loop

IF scores[j] >= best – 10, output “Student “ + j + “ score is “ + score[j] + “ and grade is A”

ELSE IF scores[j] >= best – 10, output “Student “ + j + “ score is “ + score[j] + “ and grade is A”

ELSE IF scores[j] >= best – 20, output “Student “ + j + “ score is “ + score[j] + “ and grade is B”

ELSE IF scores[j] >= best – 30, output “Student “ + j + “ score is “ + score[j] + “ and grade is C”

ELSE IF scores[j] >= best – 40, output “Student “ + j + “ score is “ + score[j] + “ and grade is D”

ELSE output “Student “ + j + “ score is “ + score[j] + “ and grade is F”

End

Function findBestGrade( requires Double array scores, double best )

FOR EACH score in scores

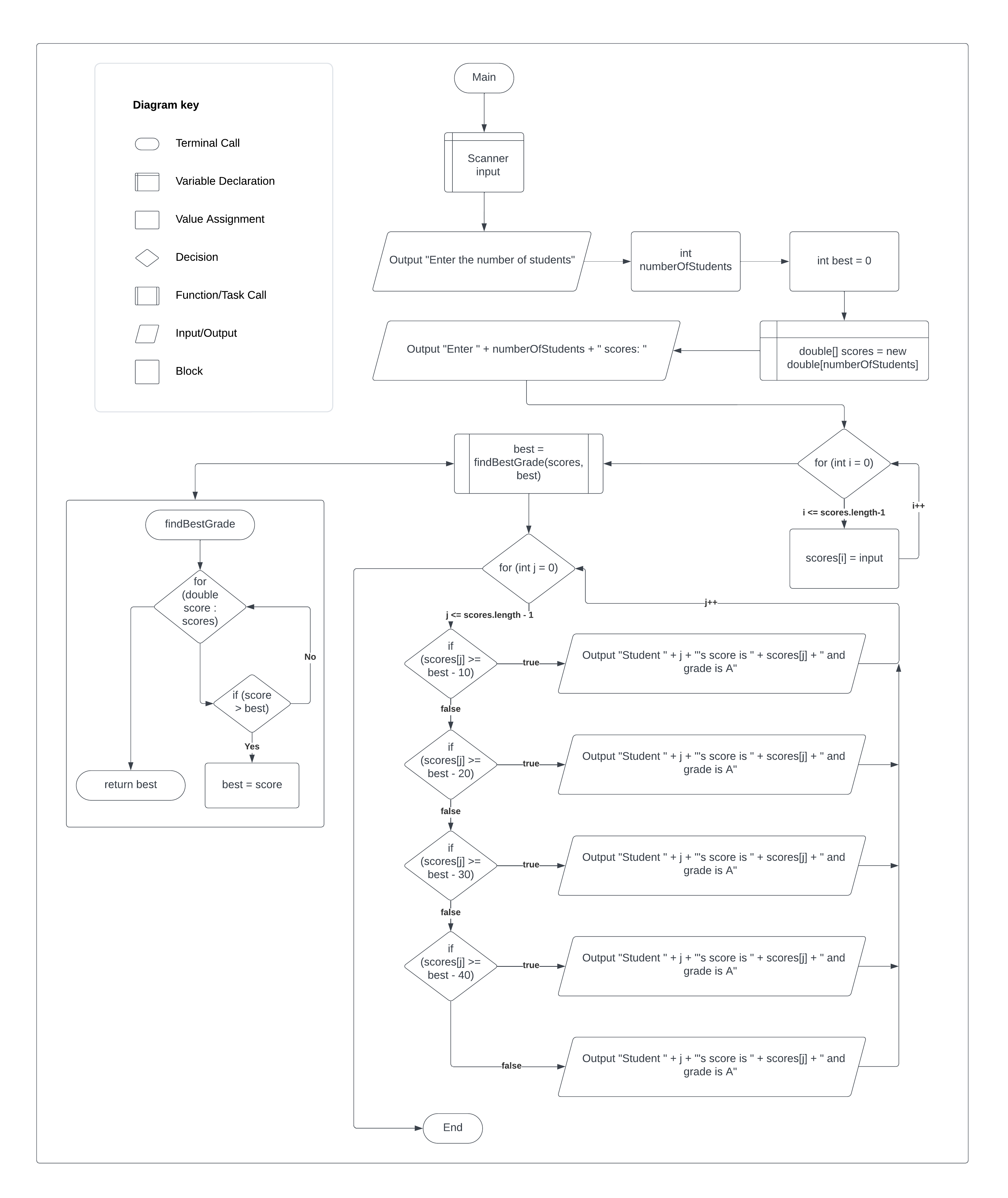
IF score is greater than best

Assign score as best

Return best

End

Flowchart



Test Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1a | Prompt user input for number of students as integer | User inputs an integer for the number of students | “Enter the number of students: “  Int numberOfStudents = input.nextInt(); | Pass |
| 2a | Prompt user input for the scores for the input number of students | User inputs as many scores as there are students | “Enter “ + numberOfStudents + “ scores:” | Pass |
| 3a | For each input double, assign input to scores array | User’s input gets stored in double array named scores | For (int I = 0; I <= scores.length – 1; i++) { scores[i] = input.nextDouble(); } | Pass |
| 4a | Pass scores and best to method findBestGrade | Double array scores and double best are passed to method findBestGrade | Best = findBestGrade(scores, best); | Pass |
| 5a | Iterate through scores to find the best score. Return best score | For each score in scores, if score is greater than best, assign score as best and return best to method main | For (double score : scores) { if (score > best) { best = score } }  Return best; | Pass |
| 6a | Iterate through scores and output each student, the score, and the letter grade | Output “Student “ x “’s score is “ scores[j] “ and grade is “ based off of the score versus the best score | For (int j = 0; j <= scores.length – 1; j++) { if (scores[j] >= best – 10) { System.out.println(“Student” + j + “’s score is “ + scores[j] + “ and grade is A”);  Etc.. | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input/Output** | **Expected Result** | **Actual Result** | **Outcome (Pass/Fail)** |
| 1 | (1a) 4  (2a) 40 55 70 58 | (6a) Student 0’s score is 40.0 and grade is C  Student 1’s score is 55.0 and grade is B  Student 2’s score is 70.0 and grade is A  Student 3’s score is 58.0 and grade is B | (6a) Student 0’s score is 40.0 and grade is C  Student 1’s score is 55.0 and grade is B  Student 2’s score is 70.0 and grade is A  Student 3’s score is 58.0 and grade is B | Pass |
| 2 | (1a) 5  (2a) 60 90 86 92 40 | (6a) Student 0’s score is 60.0 and grade is D  Student 1’s score is 90.0 and grade is A  Student 2’s score is 86.0 and grade is A  Student 3’s score is 92.0 and grade is A  Student 4’s score is 40 and grade is F | (6a) Student 0’s score is 60.0 and grade is D  Student 1’s score is 90.0 and grade is A  Student 2’s score is 86.0 and grade is A  Student 3’s score is 92.0 and grade is A  Student 4’s score is 40 and grade is F | Pass |
| 3 | (1a) 3  (2a) 100 80 50 | (6a) Student 0’s score is 100.0 and grade is A  Student 1’s score is 80.0 and grade is B  Student 2’s score is 50.0 and grade is F | (6a) Student 0’s score is 100.0 and grade is A  Student 1’s score is 80.0 and grade is B  Student 2’s score is 50.0 and grade is F | Pass |
| 4 | (1a) 8  (2a) 40 45 50 55 60 65 70 75 | (6a) Student 0’s score is 40.0 and grade is D  Student 1’s score is 45.0 and grade is C  Student 2’s score is 50.0 and grade is C  Student 3’s score is 55.0 and grade is B  Student 4’s score is 60.0 and grade is B  Student 5’s score is 65.0 and grade is A  Student 6’s score is 70.0 and grade is A  Student 7’s score is 75.0 and grade is A | (6a) Student 0’s score is 40.0 and grade is D  Student 1’s score is 45.0 and grade is C  Student 2’s score is 50.0 and grade is C  Student 3’s score is 55.0 and grade is B  Student 4’s score is 60.0 and grade is B  Student 5’s score is 65.0 and grade is A  Student 6’s score is 70.0 and grade is A  Student 7’s score is 75.0 and grade is A | Pass |

A screenshot of a computer program

Description automatically generated